

**CHARACTERIZATION OF HVAC OPERATION UNCERTAINTY  
IN ENERGYPLUS AHU MODULES**

A Thesis  
Presented to  
The Academic Faculty

by

DI SUI

In Partial Fulfillment  
of the Requirements for the Degree  
Master of Science in the  
School of Architecture

Georgia Institute of Technology  
May 2014

[Copyright 2014 by Di Sui]

**CHARACTERIZATION OF HVAC OPERATION UNCERTAINTY  
IN ENERGYPLUS AHU MODULES**

Approved by:

Professor. Godfried Augenbroe, Advisor  
School of Architecture  
*Georgia Institute of Technology*

Dr. Jason Brown  
School of Architecture  
*Georgia Institute of Technology*

Date Approved: 6 April 2014

## **ACKNOWLEDGEMENTS**

I would like to express my very most sincere gratitude to my advisor, Professor Godfried Augenbroe, for sparing his time so generously in giving me visionary guidance during the planning and development of this research. His enthusiasm in research also ignite my interest in study, which supports me over my whole degree program.

I would also like to offer my special thanks to Yuming Sun, who figures out a method to quantify the uncertainties, helps me code it into EnergyPlus, and always provides constructive solutions to my problems. This thesis could not be accomplished without his considerable supports.

Specially thanks to Qi Li, Yuna Zhang and QinPeng Wang, who devoted valuable time on discussions of this research, and to EnergyPlus Helpdesk working team, who offered me with the most authoritative answers to my questions. I would like to thank them for always being there when I need help.

Last but not least, I would like to give my deep appreciation and best love to my parents, my boyfriend, and my friends, who always stood by my side during the whole research. I would like to thank them for always believe in me, and give my the courage to keep fighting.

## TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS .....	III
LIST OF TABLES .....	VI
LIST OF FIGURES .....	VII
SUMMARY .....	XI
Chapter 1 INTRODUCTION .....	1
Chapter 2 BUILDING MODEL DESCRIPTION .....	4
Chapter 3 ENERGY PLUS SYSTEM CALLING .....	7
Chapter 4 SELECTED UNCERTAINTY PARAMETERS .....	9
SYSTEM UQ: OA CONTROLLER .....	10
SYSTEM UQ: HEATING COIL UA .....	13
SYSTEM UQ: TEMPERATURE CONTROLLER .....	15
SYSTEM UQ: HOT WATER DISTRIBUTION IN AIR HANDLING UNIT .....	18
SYSTEM UQ: AIR DISTRIBUTION .....	27
SYSTEM UQ: WATER DISTRIBUTION IN VAV REHEAT COILS .....	50
Chapter 5 SIMULATION RESULTS ANALYSIS .....	58
HEATING COIL UA FACTOR UNCERTAINTY .....	58
TEMPERATURE SENSOR UNCERTAINTY .....	62
OA CONTROLLER UNCERTAINTY .....	63
WATER AND AIR DISTRIBUTION .....	71

Chapter 6 CONCLUSION AND FUTURE WORK .....	78
REFERENCES .....	81

## LIST OF TABLES

	Page
Table 1 Daily Average Energy Consumption with UQ: Heating Coil UA	61
Table 2 Daily Average Energy Consumption with UA: OA Controller, January	65
Table 3 Daily Average Energy Consumption with UA: OA Controller, July	69
Table 4 Daily Average Energy Consumption with UA: Water and Air Distribution, January	73
Table 5 Daily Average Energy Consumption with UA: Water and Air Distribution, July	77
Table 6 Impact of each uncertainty to hourly, daily and monthly energy consumptions	78

## LIST OF FIGURES

	Page
Figure 1 Reference Building Model Upper Level (AHU1,2,3) Air Loop diagram	5
Figure 2 Reference Building Model Basement (AHU5) Air Loop diagram	5
Figure 3 Reference Building Model Hydronic System Loop Diagram	6
Figure 4 EnergyPlus Top Level Calling Tree	7
Figure 5 High Level HVAC Calling Tree (schematic-not all routines are shown)	8
Figure 6 Air Flow Network	11
Figure 7 Added Module in IDF File for UQ: OA Controller	12
Figure 8 Working Process of Subroutine SimulateWaterCoilComponents	14
Figure 9 Added Module in IDF File for UQ: Heating Coil UA	15
Figure 10 Controller: Water Coil	16
Figure 11 Added Module in IDF File for UQ: Temperature Controller	17
Figure 12 Working Process of Subroutine: SolveAirLoopControllers	23
Figure 13 Working Process of Subroutine: ManageControllers	23
Figure 14 Working Process of the updated Subroutine: SolveAirLoopControllers	25
Figure 15 Working Process of Subroutine: ReUpdateController	25
Figure 16 Added Module in IDF File for UQ: Hot Water Distribution	26
Figure 17 Added Module in IDF File for UQ: Chilled Water Distribution	26
Figure 18 Working Process of Subroutine: ManageZoneEquipment	28
Figure 19 Working Process of Subroutine: SimVAV	28
Figure 20 Zone Equipment System Simulation Calling Tree	34

Figure 21 updated Working Process of Subroutine: ManageZoneEquipment	49
Figure 22 Updated Zone Equipment System Simulation Calling Tree	49
Figure 23 Added Module in IDF File for UQ: Air Distribution	50
Figure 24 Working Process of Subroutine SimVAV with Simulation of Reheat Coils Marked in Red	51
Figure 25 Working Process of Subroutine: ControlCompOutput	53
Figure 26 Hourly Energy Consumption Difference between Systems with and without Heating Coil UA Uncertainty	59
Figure 27 Hourly Energy Consumption and Hourly Average Energy Consumption Difference Distributions for UQ: Heating Coil UA	59
Figure 28 Hourly Max and Min Energy Consumption Difference Analysis for UQ: Heating Coil	60
Figure 29 Daily Max and Min Energy Consumption Difference Analysis for UQ: Heating Coil UA	61
Figure 30 Monthly Energy Consumption Difference Distribution over 50 Samples for System UQ: Heating Coil UA	61
Figure 31 Hourly Energy Consumption Differences between Systems with and without System UQ: OA Controller, January	63
Figure 32 Monthly Energy Consumption Difference Distribution over 50 Samples for System UQ: OA Controller, January	64
Figure 33 Hourly Max and Min Energy Consumption Difference Analysis for UQ: OA Controller, January	64



Figure 34 Daily Max and Min Energy Consumption Difference Analysis for UQ: OA Controller, January	65
Figure 35 Hourly Energy Consumption and Hourly Average Energy Consumption Difference Distributions for UQ: OA Controller, January	66
Figure 36 Hourly Energy Consumption Difference for January 4 <sup>th</sup>	67
Figure 37 Hourly Energy Consumption Difference for January 11 <sup>th</sup>	67
Figure 38 Hourly Energy Consumption Difference for January 18 <sup>th</sup>	67
Figure 39 Hourly Energy Consumption Difference for January 25 <sup>th</sup>	67
Figure 40 Hourly Energy Consumption Differences between Systems with and without System UQ: OA Controller, July	68
Figure 41 Monthly Energy Consumption Difference Distribution over 50 Samples for System UQ: OA Controller, July	68
Figure 42 Hourly Max and Min Energy Consumption Difference Analysis for UQ: OA Controller, July	69
Figure 43 Daily Max and Min Energy Consumption Difference Analysis for UQ: OA Controller, July	69
Figure 44 Hourly Energy Consumption and Hourly Average Energy Consumption Difference Distributions for UQ: OA Controller, July	70
Figure 45 Hourly Energy Consumption Differences between Systems with and without System UQ: Water and Air Distribution, January	71
Figure 46 Monthly Energy Consumption Difference Distribution over 50 Samples for System UQ: Water and Air Distribution, January	72

Figure 47 Hourly Max and Min Energy Consumption Difference Analysis for UQ: Water and Air distribution, January	72
Figure 48 Daily Max and Min Energy Consumption Difference Analysis for UQ: Water and Air Distribution, January	73
Figure 49 Hourly Energy Consumption and Hourly Average Energy Consumption Difference Distributions for UQ: Water and Air Distribution, January	74
Figure 50 Hourly Energy Consumption Differences between Systems with and without System UQ: Water and Air Distribution, July	75
Figure 51 Monthly Energy Consumption Difference Distribution over 50 Samples for System UQ: Water and Air Distribution, July	75
Figure 52 Hourly Max and Min Energy Consumption Difference Analysis for UQ: Water and Air Distribution, July	76
Figure 53 Daily Max and Min Energy Consumption Difference Analysis for UQ: Water and Air Distribution, July	76
Figure 54 Hourly Energy Consumption and Hourly Average Energy Consumption Difference Distributions for UQ: Water and Air Distribution, July	77

## SUMMARY

This study addresses 5 uncertainties that exist in the operation of HVAC systems, which will presumably affect the actual energy consumption of the HVAC system in comparison to the consumption under idealized behavior. We assume that the real operation deviates from the ideal (as-designed) operation as the result of uncertainties in the real (in contrast to ideal) physical state and operation of the components of the HVAC system. The main objective is to make it possible to capture the deviation in a building energy simulation. In this study we limit ourselves to one type of system with the main aim to test our approach. Hence we use a DOE reference model for a large office building located in Atlanta to focus the development on a particular type of HVAC system: air based heating and cooling by air handling unit (AHU) with VAV terminal boxes, using boiler and chiller as plant, as it is the most widely used system type in the US. The study is based on building energy simulation with EnergyPlus which represents the ideal operation case, i.e. the system operates as directed by the modeler and implemented in the standard EnergyPlus component modules. By investigating the source code of EnergyPlus, we locate the parameters that can express the 5 types of system operation uncertainty. We consequently add these parameters and their uncertainty range into the source code, eventually resulting in an EnergyPlus program in which the HVAC operation uncertainty is embedded as so-called model form uncertainty. The upgraded EnergyPlus is tested for each parameter uncertainty separately, and to show the impact of each uncertainty albeit for hypothetical uncertainty ranges of the parameters. The results of these tests prove to be promising.

Future work will apply the theoretical results of this study in practical building simulation, details of which are addressed towards the end of this thesis.

## **CHAPTER 1**

### **INTRODUCTION**

Building energy simulation tools have been widely used to facilitate design for new constructions and support evaluation and renovation for existing buildings. However, significant deviations in terms of building energy consumption can be observed between model-predicted results and building meter data. Deviations in energy consumption between predicted and actual building can be attributed to simulation uncertainties introduced by three components in our predictions: the accuracy of the underlying models in simulation tools, the actual weather, and the actual building usage scenarios (Wang et.al., 2012). To get a better energy simulation result, those deviations should be captured and taken into consideration. In this study we focus on how to improve the accuracy of simulation models.

As one major part of a building, HVAC system is considered as an important simulation object in building energy modeling since it is the bridge of converting “the amount of energy required to keep the building performing in a comfortable way” to “the amount of source energy that systems consume for running the building”. Advanced simulation tools such as EnergyPlus represent HVAC systems as truthfully as possible, taking all important parameters into consideration. Inevitably, energy models assume that every part of the system works as directed by the schedule and under ideal conditions. This is obviously not the case in reality, and deviations between ideal and actual operation will contribute to the performance gap, i.e. the difference between simulated and measured energy consumption. Simulation results can become more reliable if all deviations between ideal and actual behavior can be predicted and taken into consideration.

Studies have been carried out to capture those deviations using different viewpoints. One viewpoint is to directly capture in a non-intrusive way the differences observed in real buildings. Typically one can build a Gaussian Process model with measured training data, then use the model with new inputs to make predictions about uncertainty in outcomes (Yan & Malkawi, 2012). Despite that this method doesn't require the model itself to be accurate, it can only relatively well predict the building energy consumption on condition that groups of reliable training data are accessible. In our study we are interested in the prediction at the design stage. In that case we can only rely on intrusive methods of capturing the uncertainties at the model level. Therefore detailed causes of deviations should be embedded in module modifications in the HVAC system modules of the simulation software. In our case we do that through the introduction of 5 specific parameters that capture uncertainty in the AHU operation.

The simulation tool that we use for our analysis is EnergyPlus, which is one of the most widely used building performance simulation tools. HVAC systems can deviate from their intended behavior by two causes that we will briefly describe. Building equipment fails or has faulty operation over its life span, like a temperature sensor that suddenly fails, or drifting over time. We called this kind of cause "fault" (M. Basarkar et.al., 2011). Another cause is that even when the building equipment is new and operating normally, most of the time it will not perform exactly as ideally expected. For instance, a damper may open to 91% or 89% when it is told to open 90%. We call this type of cause "operation uncertainty". This paper only focuses on the second cause, i.e. uncertainties in HVAC system operation.

To make it possible to add operation uncertainties as a part of the model in a simulation, detailed possible causes of uncertainties are identified and added into EnergyPlus source code to change the values of corresponding system parameters. The

EnergyPlus HVAC model used in this study and the added uncertainty parameters will be explained in the following sections.

## **CHAPTER 2**

### **BUILDING MODEL DESCRIPTION**

The DOE commercial building reference model in compliance with ASHRAE 90.1-2010 for a large office building located in Atlanta is used as the baseline for the study. The building is mainly served by only one type of HVAC system: air based heating and cooling by air handling unit (AHU) with VAV terminal boxes, using boiler and chiller as plant. The reason why we choose this type of system is because that it is the most widely used one for commercial buildings.

The building has four stories with one basement and three upper floors. one thermal zone for the basement as a whole, and 15 thermal zones for three upper levels, each level has 5 zones (four perimeter zones and one core zone).

Each floor is served by one variable volume AHU with VAV reheat terminal boxes in the zones. In each AHU there's one heating coil and one cooling coil. Each VAV terminal box has one reheat coil. In total there are 20 heating coils in the system. Each AHU also has an OA damper to bring in outdoor air controlled by an economizer.

The building has one boiler that serves hydronic hot water to the building, and two chillers which together provide chilled water.

The major system diagrams of the model are shown as below.

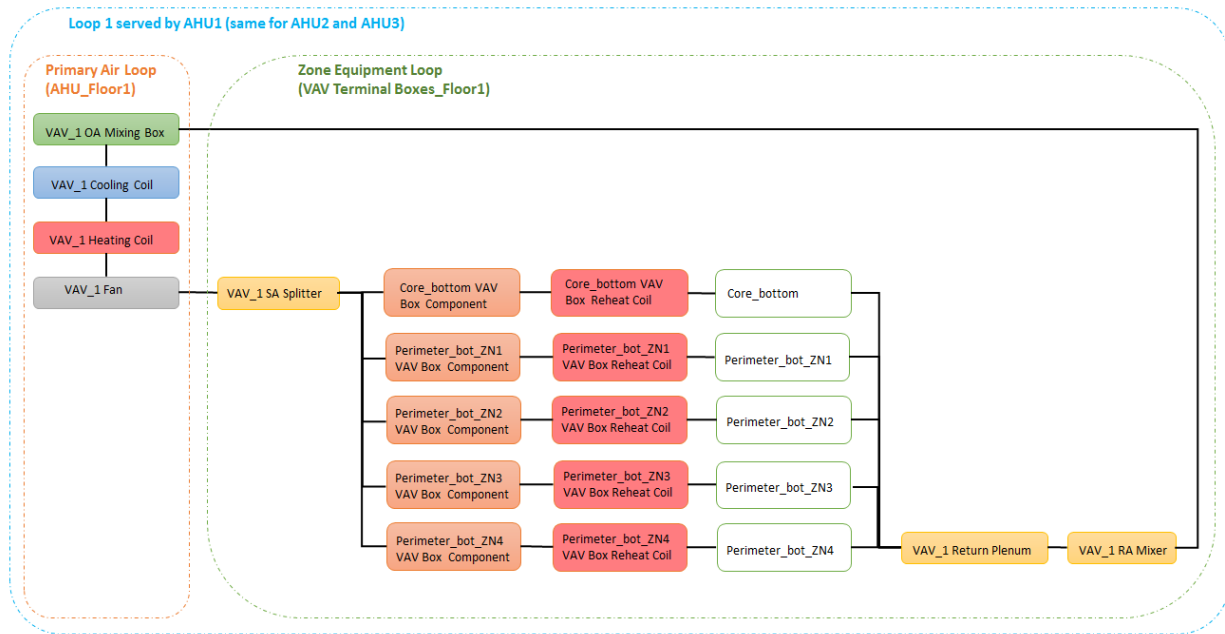


Figure 1 Reference Building Model Upper Level (AHU1,2,3) Air Loop diagram

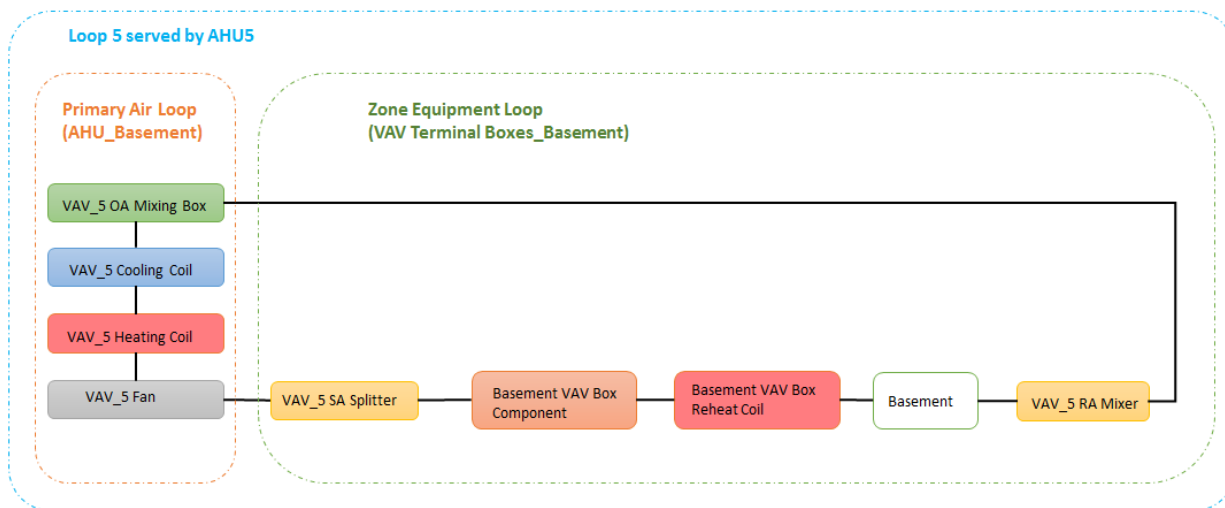


Figure 2 Reference Building Model Basement (AHU5) Air Loop diagram



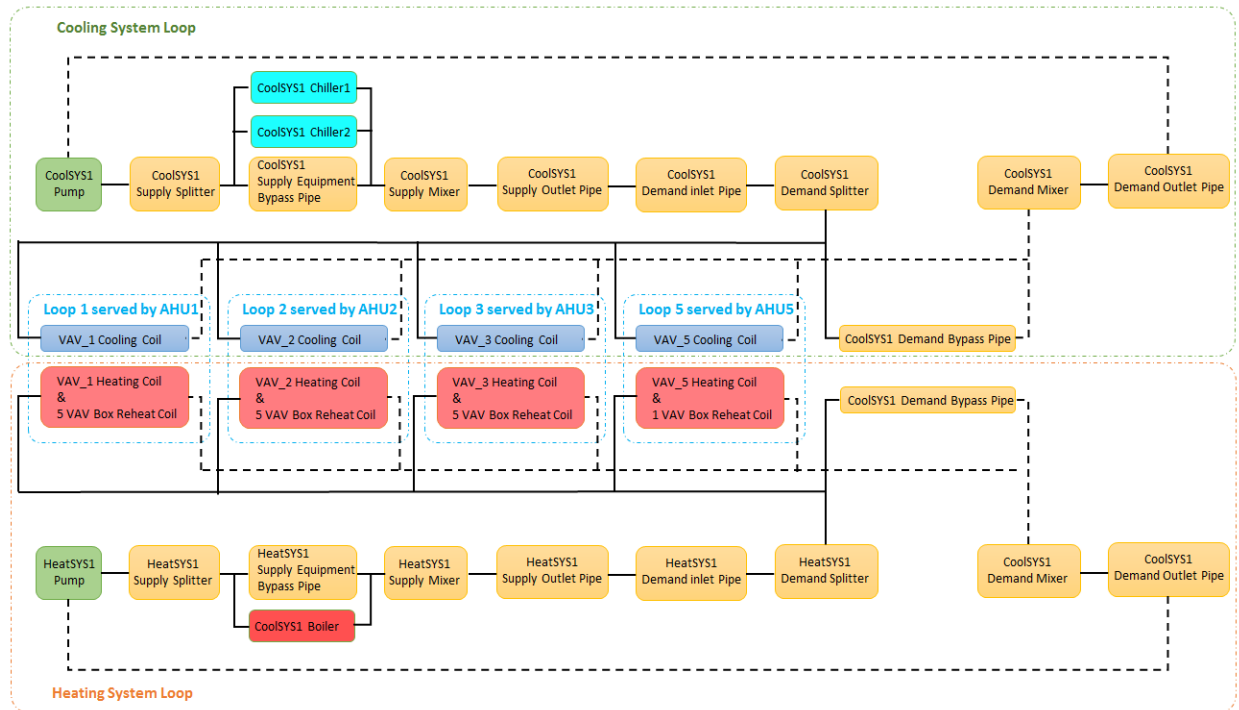


Figure 3 Reference Building Model Hyronic System Loop Diagram

## CHAPTER 3

### ENERGY PLUS SYSTEM CALLING

Following is the EnergyPlus top level calling tree. Based on *EnergyPlus Model Developer*<sup>1</sup>, EnergyPlus deals with the simulation conditions first before getting into HVAC simulation.

#### Top Level Calling Tree

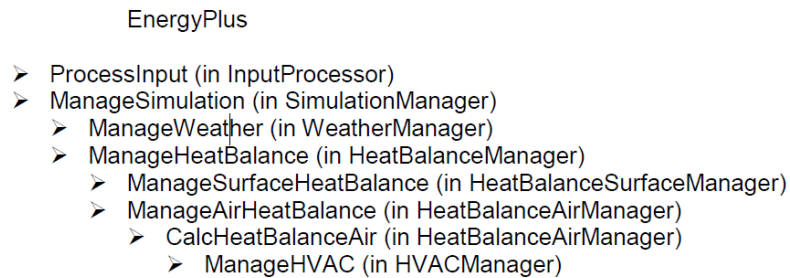


Figure 4 EnergyPlus Top Level Calling Tree

The HVAC part of EnergyPlus is divided into a number of simulation blocks. For our reference building model, there are 3 blocks: the air system, the zone equipment, and the plant equipment. Within each HVAC time step, the blocks are simulated repeatedly until the conditions on each side of each block interface match up (EnergyPlus, 2011). The following calling tree represents the schematic high level HVAC simulation structure (not all routines are shown).

---

<sup>1</sup> EnergyPlus helping documentation

### High Level HVAC Calling Tree (schematic – not all routines are shown)

ManageHVAC (in HVACManager)

- ZoneAirUpdate('PREDICT', . . .) (in HVACManager)  
*estimate the zone heating or cooling demand*
- SimHVAC (in HVACManager)
  - ManageSetPoints (in SetPointManager)
  - SimSelectedEquipment (in HVACManager)
    - ManageAirLoops (in SimAirServingZones)
    - ManageZoneEquipment (in ZoneEquipmentManager)
    - ManageElectricLoadCenters (in ElectricPowerManager)
    - ManagePlantLoops (in PlantManager)
- ZoneAirUpdate('CORRECT', . . .) (in HVACManager)  
*From the amount of heating and cooling actually provided by the HVAC system, calculate the zone temperatures.*

Figure 5 High Level HVAC Calling Tree (schematic-not all routines are shown)

## **CHAPTER 4**

### **SELECTED UNCERTAINTY PARAMETERS**

There is no prior work on encapsulating operation uncertainty in HVAC components at the level of the empirical models in EnergyPlus. Nor is there a theory that would let us identify the minimal and useful set of parameters that would sufficiently capture the deviations from ideal behavior. This section postulates a number of parameters based on the reasoning that certain physical states of the AHU are hard to control in reality. Each of these physical states is then reduced to a postulated parameter that can be positioned in the EnergyPlus modules that make up the AHU. This process is based on sound reasoning and intuition. The following sections will make this clear.

As shown in reference building model, the HVAC system consists of three major parts: 1) primary air loop which includes heating and cooling coils in AHU, outdoor air controller and mixture, and supply air fan. 2) Zone equipment loop which includes the zone VAV boxes and associated components. 3) the plant loop which includes chiller, boiler and other associated components. The uncertainties we detected are from the first and second part, located in the major components of the system.

The 5 uncertainties thus identified are: heating coil UA factor uncertainty located in heating coils, OA controller uncertainty located in outdoor air controller, temperature sensor uncertainty located at the outlet of heating coil in AHU, water distribution uncertainty located in the distribution of heating and cooling water that supplied into coils, and air distribution uncertainty located in the distribution of supply air. Each of these will be treated in detail below.

## System UQ: OA Controller

### ***Causes:***

In real system, the mass flow rate of outdoor air (OA) is calculated based on system requirements and environment conditions. This result will then be converted into a damper opening signal and passed to the OA damper. As we all know, most dampers are not accurate enough to bring in the exact amount of air we expect in the ideal case. However, this damper uncertainty has not been taken into consideration in the current EnergyPlus version.

### ***Description:***

The calculation of outdoor air is performed by a subroutine called “manage outside air system”, which is the first part of the air system block “Manage air loop”. This subroutine mainly contains two steps. The first step is called “simulate OA Controller”, which calculates the outdoor air mass flow rate based on system settings (whether the system is using demand control ventilation or minimum outdoor air mass flow rate), and calculates the relief air (RlFA) mass flow rate based on outdoor air mass flow rate and exhaust air mass flow rate (EA). The second step is called “simulate OA component”. In this step, all components in the outdoor air system are simulated in physical order. Components include OA Mixer, Water Coil, Heat Recovery, Dehumidifier, and OA Fan. OA Mixer is the first component which gets values from OA Controller, calculates the mass flow rate of recycle air (RcyA), mixes it with outdoor air, and provides the mixed air conditions for further calculations in the following components.

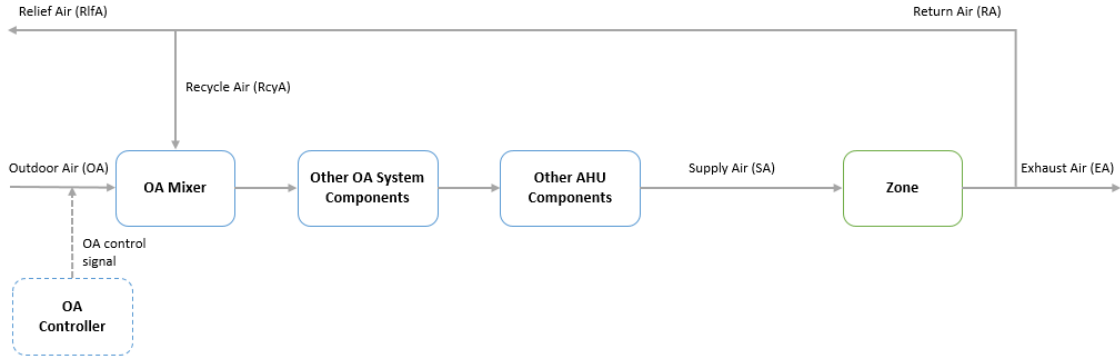


Figure 6 Air Flow Network

OA Controller:  $\dot{m}_{RlfA} = \dot{m}_{OA} - \dot{m}_{EA}$

OA Mixer:  $\dot{m}_{RcyA} = \dot{m}_{RA} - \dot{m}_{RlfA}$

In order to put the damper uncertainty into EnergyPlus, two tasks should be solved:

1. Where to insert the uncertainty
2. How to model the uncertainty

For task 1, we changed the OA mass flow rate directly after it is calculated by OA controller in “simulate OA Controller”, right before relief air mass flow rate is calculated. In this way, the new outdoor air mass flow rate will impact the relief air mass flow rate, and keep the total mixed air mass flow rate unchanged in OA Mixer.

For task 2, we define a new parameter as “OADamperUQValue”, and calculated the new OA mass flow rate as below:

$$\begin{aligned} & \text{OAController}\% \text{OAMassflow} \\ &= (1 + \mathbf{OADamperUQValue}) \times \text{OAController}\% \text{OAMassflow} \end{aligned}$$

This equation was not written in the original module, instead, we wrote it in a new UQ module. During each execution of the OA Controller, this UQ module will be called to update the OA mass flow rate at the spot described above in task 1.

Before the update, the UQ module will have to generate an **OADamperUQValue** first. The **OADamperUQValue** should vary in every time step, and from damper to

damper. However, for each damper, we considered it to have a certain range for all its **OADamperUQValue**. The uncertainty value, however varies in each time step, always stay in the range. Thus, the **OADamperUQValue** will be generated based on uniform distribution with a pair of symmetric upper and lower bounds which is a user input (we use +/- 10% for our simulation).

If the generated value is positive, the OA controller will pass a larger OA mass flow rate value then the ideal one (original result) to the OA Mixer; If negative, the final output of the OA Controller will be smaller then it's original result.

### ***Implements in IDF file***

An IDF file is an EnergyPlus input file generated by user. It contains all the information that will be used for building simulation. In order to add OA controller uncertainty into EnergyPlus, a new module will be added and shows in IDF file for user to enter the upper bound, lower bound and the uniform distribution seeds in order to generate **OADamperUQValue**. The screenshot of the module in IDF is shown as below.

Class List

Uncertainty Quantification: HVAC System

[0001] SystemUQ:SystemUQControl

[.....] SystemUQ:ChillerElectricEIR

[.....] SystemUQ:VariableSpeedPump

[.....] SystemUQ:VariableVolumeFan

[.....] SystemUQ:VariableSpeedCoolingTower

[.....] SystemUQ:HeatingCoilUA

[.....] SystemUQ:TemperatureController

[0004] SystemUQ:OAController

[.....] SystemUQ:ChilledWaterDistributionNetwork

[.....] SystemUQ:HotWaterDistributionNetwork

[.....] SystemUQ:AirDistributionNetwork:VAVReheat

Comments from IDF

Explanation of Keyword

The actual OA Mass Flow Rate Deviates from idealized Calculated Value

$OAController\%OAMassflow = (1 + OADamperUQValue) * OAController\%OAMassflow$

OADamperUQValue is within +-Maximum UQ Flow Rate Ratio Uniform distributed

ID: N4

Field	Units	Obj1	Obj2	Obj3	Obj4
OA Controller Name		VAV_1_OA_Control	VAV_2_OA_Control	VAV_3_OA_Control	VAV_5_OA_Control
OAControllerUQ SEED 1		1234	1234	1234	1234
OAControllerUQ SEED 2		3456	3456	3456	3456
OAControllerUQ SEED 3		5678	5678	5678	5678
Maximum UQ Flow Rate Ratio		0.2	0.2	0.2	0.2

*Figure 7 Added Module in IDF File for UQ: OA Controller*

## System UQ: Heating Coil UA

### ***Causes:***

In EnergyPlus, the object Coil: Heating: Water provides a model that uses an NTU-effectiveness model of a static heat exchanger. The model takes UA, the effectiveness, as well as inlet fluids conditions and flow rates as its inputs, and calculates the outlet fluids conditions as outputs (Zhang, 2001). In let fluids include inlet hot water and inlet air that ready to be heated up. The UA is either a user input, or calculated based on the design conditions of the heating coil. No matter how the UA is assigned, the value of this parameter is fixed and will not be changed during the whole simulation. However, in the real situation, the UA value for most coils are not the same as their labeled ones due to imprecisely manufacture, and the real UA of a coil in use may not equal to the rated value if the coil is not fully loaded.

### ***Description:***

The implement of UA is quite simple, both where and how to insert the UQ parameter.

As it is described above, the implement of UA could be performed in two possible ways:

1. Consider the uncertainty as an inherent attribute of each coil, and add a user defined parameter for the coil model as a ratio of how good this heating coil can do as a heat exchanger.
2. Consider the uncertainty as a function of the hot water flow rate, and renew the UA value in the simulation in every time step based on the function.

For now, since it's hard to figure out the relation between UA value and hot water flow rate, we use the first method and add a parameter named



**HeatingCoilUAUncertaintyFactor** in the heating coil model in IDF file for users to change. Once the uncertainty value is assigned, it will be keep constant for the rest of the simulation.

The logic of heating coil simulation is quite simple and straight forward in EnergyPlus. The Subroutine “SimulateWaterCoilComponents” will be called for simulating the heating coil. The figure below shows the process of this subroutine.

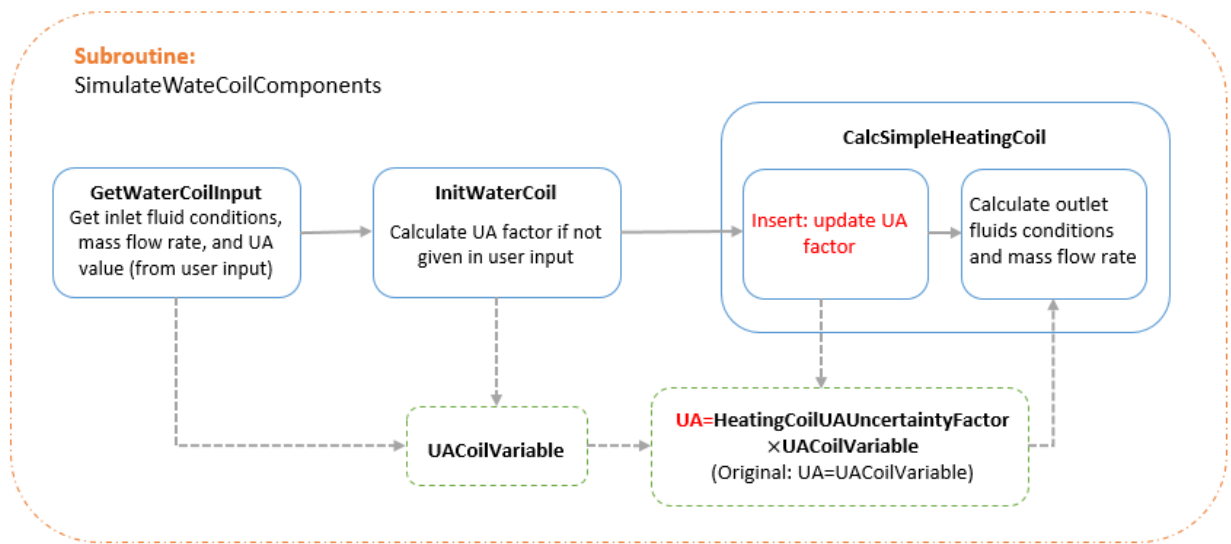


Figure 8 Working Process of Subroutine SimulateWaterCoilComponents

The uncertainty parameter **HeatingCoilUAUncertaintyFactor** will be used at the first step of the calculation.

$$UA = \text{HeatingCoilUAUncertaintyFactor} \times \text{WaterCoil\%UACoilVariable}$$

To test this uncertainty, a series of simulations will be produced using GURA-W with a group of different UA uncertainties in a range of 0.90 to 1.10 ( $\pm 10\%$ ), which will be introduced in next chapter.

### Implements in IDF file

A new module is added into IDF file for user to enter the UA uncertainty for each coil. A screenshot of the module is shown as below. Notice that the picture only shows five object, and yet there are 20 of them in the model we used.

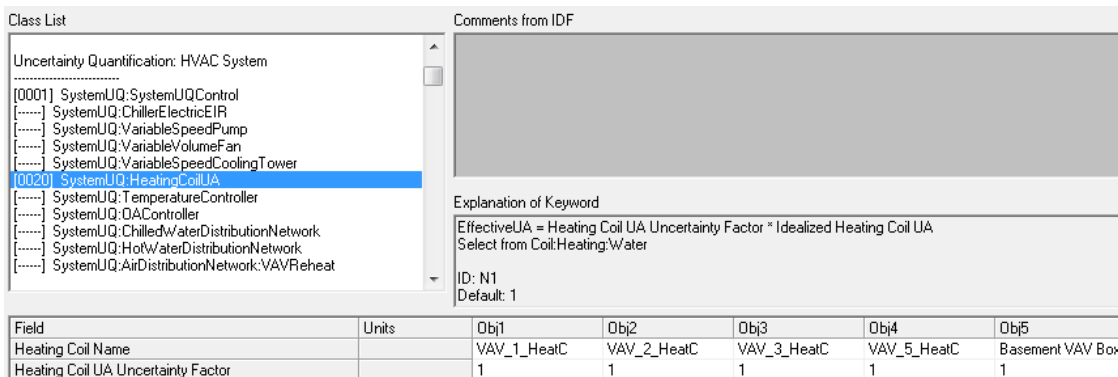


Figure 9 Added Module in IDF File for UQ: Heating Coil UA

### System UQ: Temperature Controller

#### Causes:

The inaccuracy of the air temperature sensor at the water coil outlet node.

#### Description:

Before simulating the water coil, EnergyPlus will run the corresponding water coil controller first. The figure below illustrates the principle of a mostly used simple water coil controller.

The controller reads the real outlet air temperature (sensed value) as well as the desire outlet air setpoint temperature (setpoint value) from the control node, and maintain the sensed temperature at the setpoint value by modulating the inlet water flow rate at the actuator node.

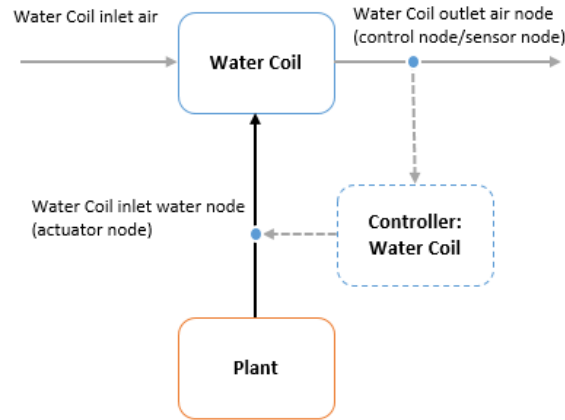


Figure 10 Controller: Water Coil

According to the principle above, EnergyPlus does the calculation below to find out the proper water flow rate (*NextActuatedValue*).

1. Read the outlet air temperature:

$$\text{SensedValue} = \text{Node}(\text{sensedNode})\% \text{Temperature}$$

2. Calculate the difference between the sensed value and setpoint value:

$$\text{DeltaSensed} = \text{SensedValue} - \text{SetpointValue}$$

3. Use root-finding algorithm to find the *NextActuatedValue* such that the corresponding *DeltaSensed* is equal to 0.

While in reality, the sensed temperature will not be the same as its actual value due to the sensor uncertainty, which will cause the HVAC Controller to come up with a different water inlet flow rate, and will impact both the plant energy consumption and the coil outlet air temperature.

For now, we consider this uncertainty as an inherent attribute of the sensor, which means the uncertainty factor will be set up only once and used for the entire simulation. (E.g. if the factor > 0, the sensor will always give a larger number than the actual value.)

To capture this accuracy of the sensor, we add a module named “Temperature Controller”. It allows user to input a value to represent the sensor inaccuracy. By changing the code of controller simulation, this user input factor will be used to add uncertainty to the *SensedValue*.

SensedValue =

Node(sensedNode)%Temperature × **TemperatureControllerUncertaintyFactor**

Same as the damper uncertainty, we will testing this sensor uncertainty using GURAW. The range of the uncertainty will be from 0.90 to 1.10 (±10%).

### ***Implements in IDF file***

A new module is added into IDF file for user to enter the temperature controller uncertainty value. A screenshot of the module is shown as below.

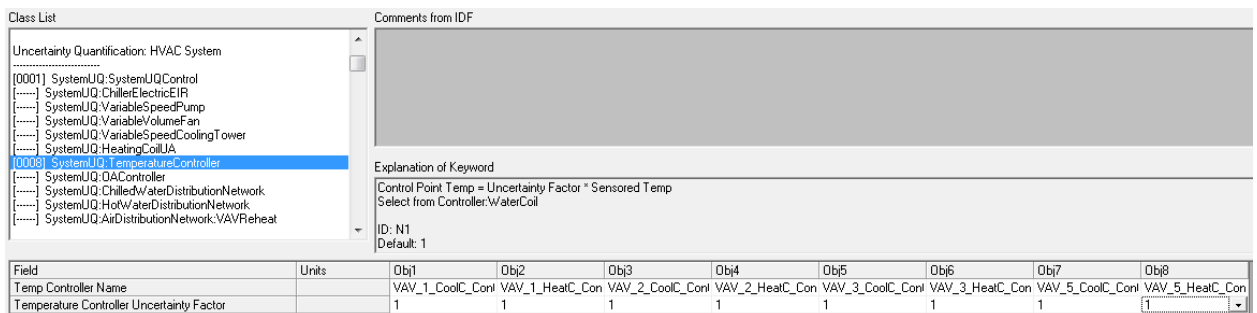


Figure 11 Added Module in IDF File for UQ: Temperature Controller

## System UQ: Hot Water Distribution in Air Handling Unit

### ***Causes:***

For each air loop (served by one AHU), there are more than one heating coils located in the air handling unit or in each VAV terminal boxes (re-heat coil). All those coils get their inlet hot water from one plant with the same pump. Thus, both the inaccuracy of the pump and the unbalance of water network will cause the uncertainty of inlet water flow rate for each heating coil.

### ***Uncertainty quantification description<sup>2</sup>:***

To capture the two types of uncertainty, let's denote:

$\dot{m}_i^d$ : Demand water mass flow rate by terminal I

$\dot{m}_i^r$ : Actual water mass flow rate received by terminal I

$\dot{M}^d = \sum_{i=1}^N \dot{m}_i^d$ : Total demand water mass flow rate by one pump

$\dot{M}^r = \sum_{i=1}^N \dot{m}_i^r$ : Total actual water mass flow rate supplied by one pump

In EnergyPlus:

$\dot{m}_i^r = \dot{m}_i^d$ , therefore,

$$\dot{M}^r = \dot{M}^d$$

---

<sup>2</sup> The uncertainty quantification method is generated by Yuming Sun and modified by Di Sui.

However, due to the pump inaccuracy, it is more realistic that

$$\dot{M}^r = (1 + \rho)\dot{M}^d$$

$\rho$ : Uncertain factor that capture the error in the use of idealized pump reaction, i.e., the actual delivered water  $\dot{M}^r$  deviates from the demand  $\dot{M}^d$ .

The follow-up question is how the extra amount of water  $\rho\dot{M}^d$  got delivered to each terminal  $i$ , namely, the unbalance problem.

Introduce another parameter  $\tau$ , which describes the dispersion of the amount of water to each terminal  $i$ .

Implementation Process:

$$X \sim N(0, \tau)$$

Generate  $N$  samples of  $(x_1, x_2, \dots, x_N)$  and ensure  $\sum x_i \leq 0.00001$  by reject invalid samples.

The sample will be fixed throughout the entire simulation run, cause once the water system is designed, the relationships between each branch are defined, and will not change if the loop structure stay the same.

$$\dot{m}_i^r = (\dot{m}_i^d + x_i \dot{M}^d)(1 + \rho)$$

If pump has no uncertainty:  $\dot{m}_i^r = \dot{m}_i^d + x_i \dot{M}^d$

If water network has no unbalancing problem:  $\dot{m}_i^r = \dot{m}_i^d(1 + \rho)$

Now we can mimic the unbalance characteristic of the water distribution system. Since  $x_i$  can have different signs, some terminals will received less water than required, while others will receive more. The total extra amount of water should equal with the extra delivered by the pump. And the law of mass conservation has to be met:

$$\begin{aligned}\sum \dot{m}_i^r &= (1 + \rho) \sum (\dot{m}_i^d + x_i \dot{M}^d) = (1 + \rho) \sum \dot{m}_i^d + (1 + \rho) \dot{M}^d \sum x_i \\ &= (1 + \rho) \dot{M}^d + 0 = \dot{M}^r\end{aligned}$$

### ***Implement in EnergyPlus***

In EnergyPlus source code, simulations for coils in air handling units and reheat coils in VAV reheat terminals are carried out in different places. The simulations for AHU coils are parts of the primary air loop simulation, which is in the section “ManageAirLoops”, while simulations for reheat coils are parts of the “ManageZoneEquipment” section. Therefore, our implementation for the water distribution uncertainty is carried out for these two types of coils separately. This section will introduce the implementation for AHU coils. The other type will be introduced after the air distribution uncertainty since they both belong to the “ManageZoneEquipment” section, and simulation of air mass flow rates comes first. The reason why those implementations are separated will be addressed after introducing the implementation for VAV reheat coils. In order to implement the uncertainties, we have to:

1. Calculate and assign  $\rho$ ,  $x_i$  to each water coil
2. Calculate total actuated water mass flow rate  $\dot{M}^d$
3. Update  $\dot{m}_i^d$  to  $\dot{m}_i^r$  using  $\dot{m}_i^r = (\dot{m}_i^d + x_i \dot{M}^d)(1 + \rho)$

The spot to insert the update in EnergyPlus source code is easy to find based on the description above.

As introduced above, the water inlet mass flow rate of each coil is calculated by the corresponding controller. The subroutine which carry out this calculation is called “CalcSimpleController”. The value of  $\dot{m}_i^d$  calculated by this subroutine is called “NextActuatedValue”. Then the value will be assigned to the corresponding water coil inlet node by a subroutine called “UpdateController”.

In “UpdateController”, *NextActuatedValue* will be assigned to two parameters:

node(water coil inlet node)% WaterFlowRateRequest = NextActuatedValue

node(water coil inlet node)% WaterFlowRate = NextActuatedValue

WaterFlowRateRequest: used for plant simulation

WaterFlowRate: used for water coil simulation

Since we want to change both of the parameters to the updated actuated value, we have to change the NextActuatedValue before it got assigned, right at the beginning of the subroutine “UpdateController”.

NextActuatedValue =

(NextActuatedValue + **WaterFlowRateUQ1** × HeatingTotalNextValue) × (1  
+ **WaterFlowRateUQ2**)

HeatingTotalNextValue:  $\dot{M}^d$

WaterFlowRateUQ1:  $x_i$

WaterFlowRateUQ2:  $\rho$



Since subroutine “UpdateController” will be used under different situations, we wrote and use a new subroutine named “ReupdateController” which contains the equation above to renew the NextActuatedValue.

The next step is to calculate and assign the WaterFlowRateUQ.

Since **WaterFlowRateUQ1** and **WaterFlowRateUQ2** should stay the same during the whole simulation, we calculated the value at the very beginning of the water flow distribution simulation, and indexed each value with corresponding water coil name.

**WaterFlowRateUQ1** is generated using a normal distribution. **WaterFlowRateUQ2** is added as a user input, and will be read from IDF file.

The last step is to figure out the HeatingTotalNextValue by summing up all the water coil inlets. Before that, we have to first figure out that in what order EnergyPlus simulates all the AHU controllers.

The simulation begins by looping over all primary air loop (One loop contains all AHU components and other mechanical equipment that doesn’t belong to plant part or zone part). Namely, EnergyPlus will simulate all the controllers (e.g. OA controller, water coil controllers for AHU coils) and all the components (fans and coils) for one primary air loop, and then move to the next one.

The simulation for each air loop is carried out by calling a subroutine called “SimAirLoop”. In this subroutine, EnergyPlus will call either “ResolveAirLoopControllers” if a warm restart is supported, or “SolveAirLoopControllers” if not. With a warm restart, “ResolveAirLoopControllers” will not call the controller to calculate a new actuated flow rate for the coil, instead, the coil will use the same value as the one in last time step. In this case, we don’t want to change the water flow rate for this coil either, thus we don’t

have to make any change to this subroutine. The illustrations of the other subroutine “SolveAirLoopControllers” is shown below.

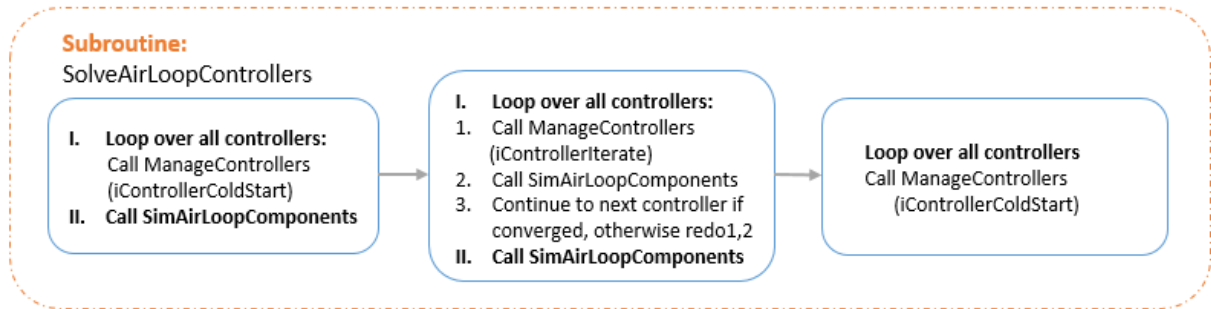


Figure 12 Working Process of Subroutine: SolveAirLoopControllers

The core subroutine called in both alternatives above is “ManageControllers”, but with different controller operation types (iControllerColdStart/ iControllerWarmRestart/ iControllerIterate/ iControllerEnd). The operation type “iControllerIterate” will call the water coil controller to calculate the actuated water flow rate, thus this is the type that we need to add uncertainty into.

The working process of “ManageControllers” with type “iControllerIterate” is shown as below.

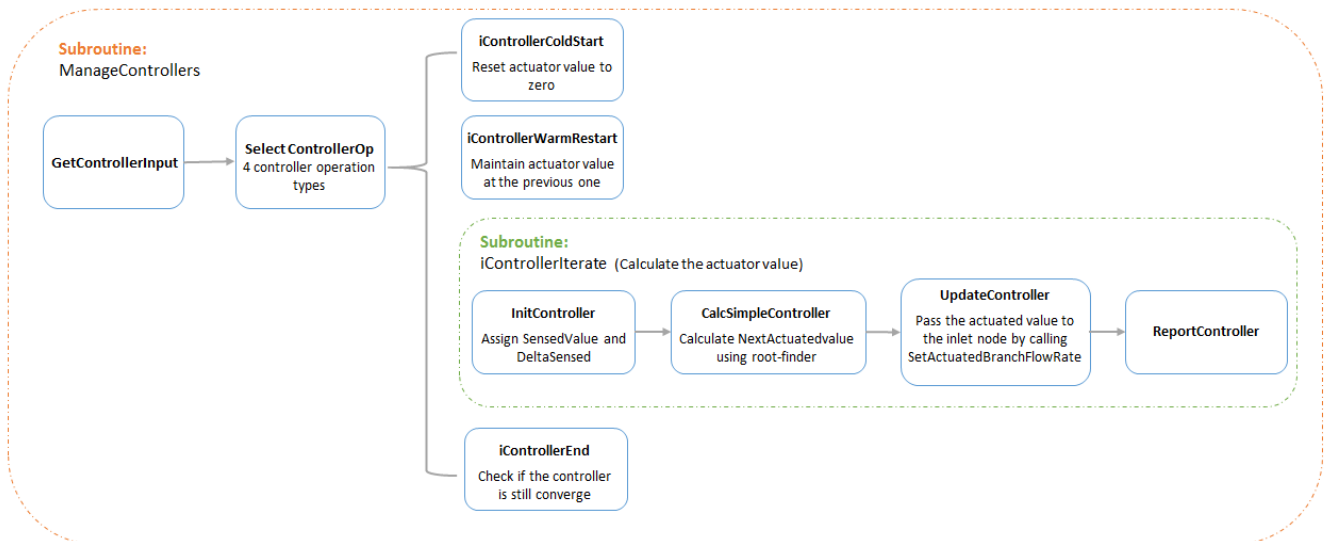


Figure 13 Working Process of Subroutine: ManageControllers

Based on the illustration above, “SolveAirLoopControllers” calls “ManageControllers”, which calls “iControllerIterate” that uses the root-finding algorithm to calculate the *NextActuatedValue*, while “ResolveAirLoopControllers” only maintain the water flow rates for the entire air loop. Thus, only “SolveAirLoopControllers” has to be changed to implement the uncertainty.

As shown in the figures above, “SolveAirLoopControllers” functions as the following three steps:

1. Initial all the controllers, reset their actuated value to 0.
2. Calls “ManageControllers” to calculate the *NextActuatedValue* in “iControllerIterate” for all the controllers using root finder;

Then calls “SetActuatedBranchFlowRate” to pass the actuated value to coil water inlet node for further simulation.

3. Simulate air components.
4. Check all the controllers to see if all are converged.

To add uncertainty to each actuated value, all the actuated value should be calculated twice. First time to get the HeatingTotalNextValue by summing up all actuated value, and second time to assign uncertainties to each actuated value using HeatingTotalNextValue and uncertainty factors.

Thus, one extra step should be added before simulate the air components (step 3) to calculate all HeatingTotalNextValue for a second time and add uncertainties into each actuated value. We name this new subroutine “UQUPdateManageControllers”.

In this new step, since all the *NextActuatedValue* for every controller in this air loop have been calculated, HeatingTotalNextValue for this air loop will be calculated by summing up all *NextActuatedValue*. Then by looping over all the controllers again with a new subroutine “UQUPdateManageController”(new subroutine written based on

“ManageController”), all the *NextActuatedValue* will be updated using uncertainty parameters. The updating process is carried out by new subroutine “ReupdateController” (witten based on “UpdateController”) called by “UQUPdateManageController”

The Figures below shows the new working process of water coil actuated flow rate calculation.

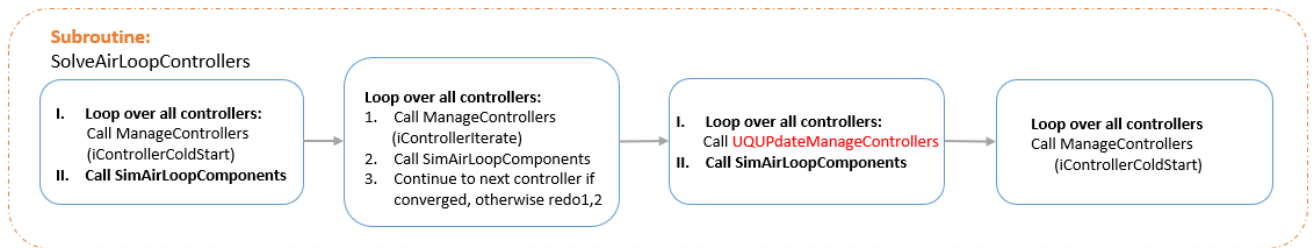


Figure 14 Working Process of the updated Subroutine: SolveAirLoopControllers

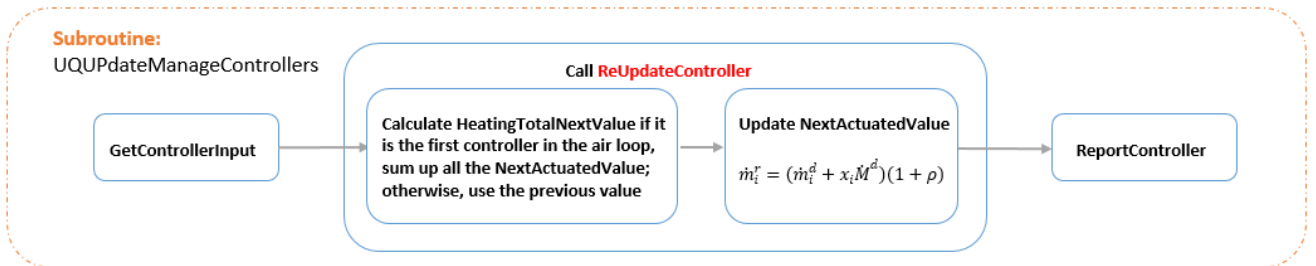


Figure 15 Working Process of Subroutine: ReUpdateController

The HeatingTotalNextValue, which is also the  $\dot{M}^d$ , is the total hot water flow rate for one primary air loop, not for all loops connected to the pump, which means that we ignore the unbalance between each air loop. The uncertainty caused by the pump for each air loop will still be  $\rho$ .

Since each air loop only has one cooling coil, there will be no unbalancing problem for cooling coils. The pump uncertainty still impacts the water flow of the cooling coils. The implementation for cooling coils are the same as for the heating coils.

### Implements in IDF file

A new module is added into IDF file for user to enter the temperature controller uncertainty value. A screenshot of the module is shown as below.

Class List

Uncertainty Quantification: HVAC System

- [0001] SystemUQ:SystemUQControl
- [.....] SystemUQ:ChillerElectricEIR
- [.....] SystemUQ:VariableSpeedPump
- [.....] SystemUQ:VariableVolumeFan
- [.....] SystemUQ:VariableSpeedCoolingTower
- [.....] SystemUQ:HeatingCoilUA
- [.....] SystemUQ:TemperatureController
- [.....] SystemUQ:OAController
- [0001] SystemUQ:ChilledWaterDistributionNetwork
- [0001] SystemUQ:HotWaterDistributionNetwork
- [0004] SystemUQ:AirDistributionNetwork:VAVReheat

Comments from IDF

Explanation of Keyword

Required total hot water deviates from delivered  
The network unbalance uncertainty

ID: N2  
Default: 1

Field	Units	Obj1
Hot Water Pump Flow Rate Control Uncertain Factor		0.1
Hot Water Distribution Network Uncertain Factor		1

Figure 16 Added Module in IDF File for UQ: Hot Water Distribution

Same as for hot water distribution in AHU, we add a module for chilled water distribution as well. The screenshot of the module is shown as below.

Class List

Uncertainty Quantification: HVAC System

- [0001] SystemUQ:SystemUQControl
- [.....] SystemUQ:ChillerElectricEIR
- [.....] SystemUQ:VariableSpeedPump
- [.....] SystemUQ:VariableVolumeFan
- [.....] SystemUQ:VariableSpeedCoolingTower
- [.....] SystemUQ:HeatingCoilUA
- [.....] SystemUQ:TemperatureController
- [.....] SystemUQ:OAController
- [0001] SystemUQ:ChilledWaterDistributionNetwork
- [0001] SystemUQ:HotWaterDistributionNetwork
- [0004] SystemUQ:AirDistributionNetwork:VAVReheat

Comments from IDF

Explanation of Keyword

Required total chilled water deviates from delivered  
The network unbalance uncertainty

ID: N2  
Default: 1

Field	Units	Obj1
Chilled Water Pump Flow Rate Control Uncertain Factor		0.1
Chilled Water Distribution Network Uncertain Factor		1

Figure 17 Added Module in IDF File for UQ: Chilled Water Distribution

## **System UQ: Air Distribution**

The causes and the quantification for uncertainty in air distribution are mainly the same as those for water distribution.

### ***Causes:***

For each air loop, the inaccuracy of fan in AHU and the unbalancing of air network will result in the deviation between the actual delivered zone air mass flow rate (in real system) and ideal zone air mass flow rate (in EnergyPlus).

### ***Uncertainty quantification description:***

Same as for water distribution

### ***Implement in EnergyPlus:***

Air distribution is simulated by zones. For each zone, EnergyPlus calculates the air mass flow rate required by each zone equipment and sum them up to calculate the total air flow rate for the supply fan as well as the total return air flow rate for the air loop. The simulation is carried out by a subroutine called "ManageZoneEquipment"

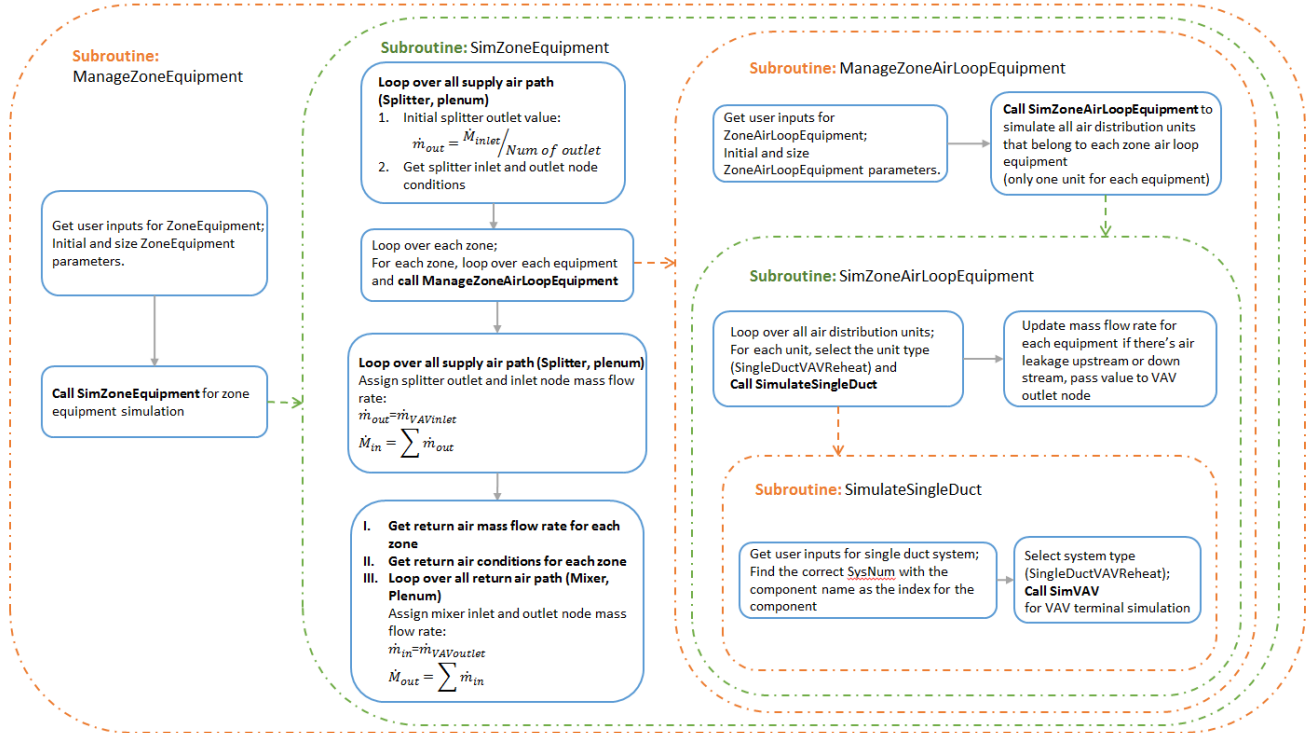


Figure 18 Working Process of Subroutine: ManageZoneEquipment

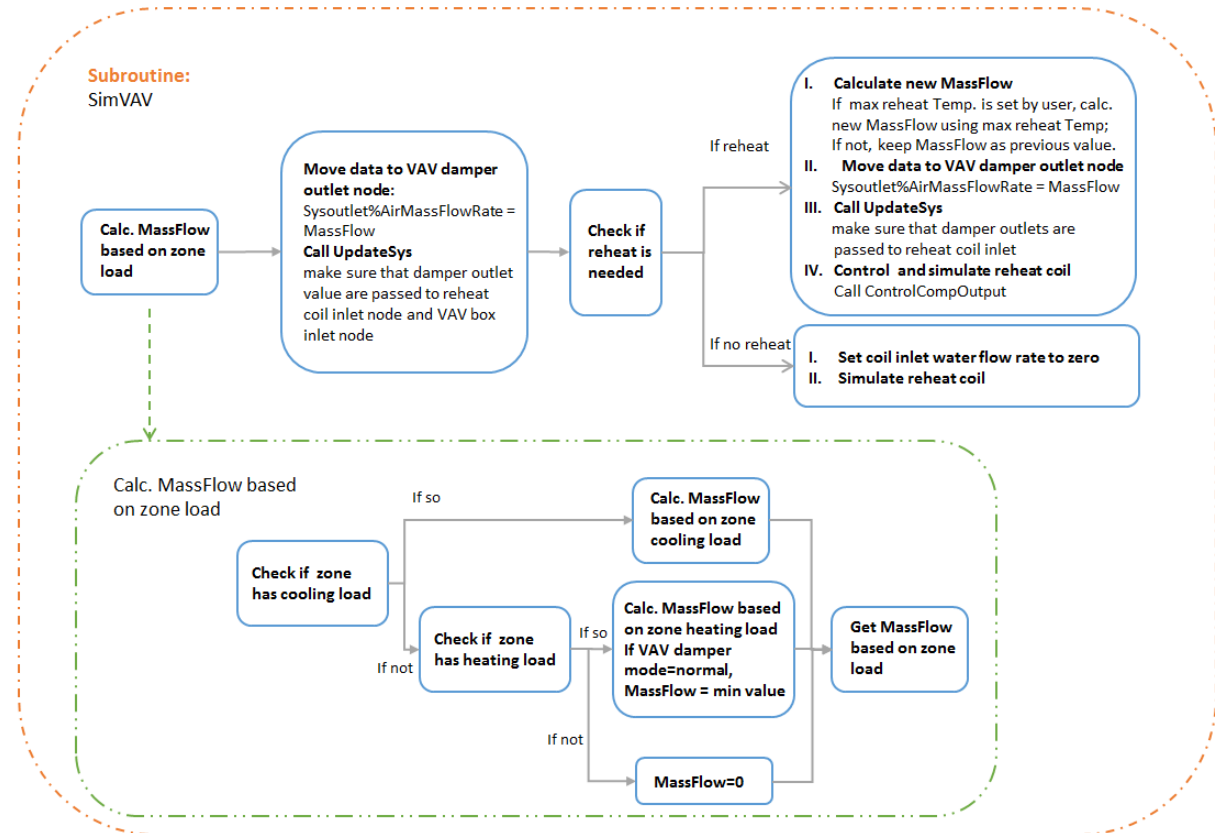


Figure 19 Working Process of Subroutine: SimVAV

As shown in Figure 18, the system air flow is calculated in three parts: supply air mass flow rate, demand air flow rate, and return air flow rate. Both supply air flow rate and return air flow rate are determined based on demand air flow rate. Thus, in order to add uncertainties to the whole system air flow, we have to make changes to the calculation of demand air flow rate.

As shown in Figure 19, the demand air mass flow rate is calculated in the subroutine “SimVAV”. In this subroutine, the value of the target air mass flow rate is always first carried by a parameter named “MassFlow”, that then immediately gets passed to VAV damper outlet node (SysOutlet), and eventually passed to the VAV reheat coil inlet node for further simulation by calling another subroutine “UpdateSys”. In order to add uncertainty to this air mass flow rate and also make sure that the value we use for further simulation is the updated one, we have to change the mass flow rate value before it got passed into the nodes.

Based on the description above, it seems that the parameter “MassFlow” is the one that should be updated since it carries the air mass flow rate value before the value got passed into the nodes. However, the calculation is carried out by trying out all the types and working conditions that this current zone VAV box could fall in, EnergyPlus will run through all the possibilities by order until it finds the value that fits the current type and working condition. The parameter “MassFlow” is a global value in this subroutine “SimVAV”, which means that the “MassFlow” will be allocated every time when one possibility is being checked, and will not be de-allocated before checking the next possibility. Thus, before making any changes, we have to first confirm that the parameter “MassFlow” under one possibility will not be used in calculation of checking any other possibilities.



Two major situation is considered in terms of type and working condition of VAV box: One is when reheat is not needed, the other one is when reheat is needed. For the second situation, the air mass flow rate will also be different based on the VAV box damper mode. According to the process illustrated in the pictures above, the air mass flow rate will be calculated and checked through 3 steps:

1. Calculate the basic air mass flow rate “MassFlow” based on zone heating/cooling load. Call “UpdateSys” to pass the value into nodes. Pay attention that if the zone has heating load and the damper mode is “normal” (during heating, air will be served in a constant flow rate, which is the damper minimum value, and change the air temperature by varying hot water flow rate in reheat coil).
2. Check if the zone needs reheat.

- a. If reheat is needed, check if the Max reheat air temperature is set by user.

- i. If Max reheat air temperature is assigned as an user input, recalculate the “MassFlow” based on the zone load, then check the “MassFlow” based on VAV damper mode:

If the mode is “normal”, which means the damper will keep providing supply air at its minimum requirement value, set “MassFlow=AirMassFlowRateMin”

If the mode is “reverse”, which means damper position will change based on zone load requirement, keep “MassFlow” as the calculated value.

- ii. If Max reheat air temperature is not assigned by user, keep “MassFlow” as the calculated value.

- b. For reheat, calculate the required hot water flow rate for reheat coil, pass the results to reheat coil inlet and outlet nodes, and simulate the reheat coil.
  - c. If the zone does not need reheat, set reheat coil hot water flow rate to zero, and simulate the reheat coil for this special case.

As shown in the description of the process above, it is clear that the parameter “MassFlow” which is calculated in the first step will be used again in the second step to determine whether the system needs reheat or not. Thus, we cannot make any changes to this parameter. Instead, the changes should be made separately for both reheat case and non-reheat case, which are at the end of step 2a and step 2c. Since we also want the updated air mass flow rate to be used in the future simulations for both return air and reheat coil, the uncertainty should be added before calling “UpdateSys”. The implements are shown as below.

At the end of step 2a:

Change:

$$\text{SysOutlet}(\text{SysNum})\% \text{AirMassFlowRate} = \text{MassFlow}$$

to:

$$\begin{aligned} \text{SysOutlet}(\text{SysNum})\% \text{AirMassFlowRate} \\ = (\text{MassFlow} + \text{Sys}(\text{SysNum})\% \text{AirFlowUQXi} \times \text{TotalAirAHU}) \\ \times (1.0d0 + \text{Sys}(\text{SysNum})\% \text{AirFlowUQRo}) \end{aligned}$$

$$\text{MassFlow} = \text{SysOutlet}(\text{SysNum})\% \text{AirMassFlowRate}$$

(Here we update MassFlow to the new value because then this parameter is used to calculate the reheat coil hot water flow rate.)

At the end of step 2c:

Change:

$$\text{SysOutlet}(\text{SysNum})\% \text{AirMassFlowRate} = \text{MassFlow}$$

to:

$$\begin{aligned}
& \text{SysOutlet}(\text{SysNum})\% \text{AirMassFlowRate} \\
& = (\text{MassFlow} + \text{Sys}(\text{SysNum})\% \text{AirFlowUQXi} \times \text{TotalAirAHU}) \\
& \times (1.0d0 + \text{Sys}(\text{SysNum})\% \text{AirFlowUQRo})
\end{aligned}$$

In the equations above, it's easy to tell that four parameters are needed in order to update the air mass flow rate: two uncertainty factors, the original air mass flow rate, and a total air mass flow rate.

#### Two uncertainty factors:

Two uncertainty factors *AirFlowUQXi* and *AirFlowUQRo* will be generated at the beginning of the simulation by calling subroutine “*GetUQInput*” and “*AirVAVDistributionUQ*”.

The methodology of generation is the same as the one for water distribution. “*GetUQInput*” reads in the AHU fan mass flow uncertain factor  $\rho$  (which is also *irFlowUQRo* ) and the air distribution network uncertain factor  $\tau$ . “*AirVAVDistributionUQ*” uses  $\tau$  to generate *AirFlowUQXi* for each zone equipment. Those uncertainty factors will be stored in an array called “*Sys*” and labeled with a number called “*SysNum*”. Later in simulation, those factors will be located by their unique “*SysNum*”, as shown in the equation above.

#### The Original air mass flow rate:

To get the original air mass flow rate “*MassFlow*”, just make sure that no change is added before the subroutine get the original value for “*MassFlow*”, as described above.

### Total air mass flow rate:

To calculate the total air mass flow rate, first we have to group the terminals.

Since VAV terminals from different AHUs have no impact on each other, and *AirFlowUQRo* for each AHU are isolated, the uncertainty has to be carried out for each AHU separately. Thus, the total air mass flow rates are calculated for each AHU separately.

In order to do so, we have to loop over all the zones twice: In loop 1, let EnergyPlus calculate air mass flow rate for each zone as it used to be, then we extract the target value at the end of the simulation for each zone, and add them up together based on their AHU numbers; in loop 2, total air mass flow rate will be to calculated the impact of the uncertainty, following the equations introduced above.

As shown in Figure 18, subroutine "*SimZoneEquipment*" is the one that carries the code of looping over all the zones. This looping-over process is carried out by a form of "Do-loops". This do-loops is the group of code that should be repeat for a second time.

In the loop1, according to equation\*, the target value of each zone terminal should be extract from parameter: *SysOutlet(SysNum)%AirMassFlowRate*. We name the target value: ***AirTerminal***. Then

$$\text{AirTerminal} = \text{SysOutlet}(\text{SysNum})\% \text{AirMassFlowRate}$$

To sum up *AirTerminal* for each AHU and use the results in loop 2, we have to pass those terminal air flow rate value to the subroutine "*SimZoneEquipment*", and calculate the total value in that at the end of loop1.

Based on Figure 18, subroutine “*SimVAV*” is not called directly by “*SimZoneEquipment*”. The calling tree between these two subroutines are shown as the figure below.

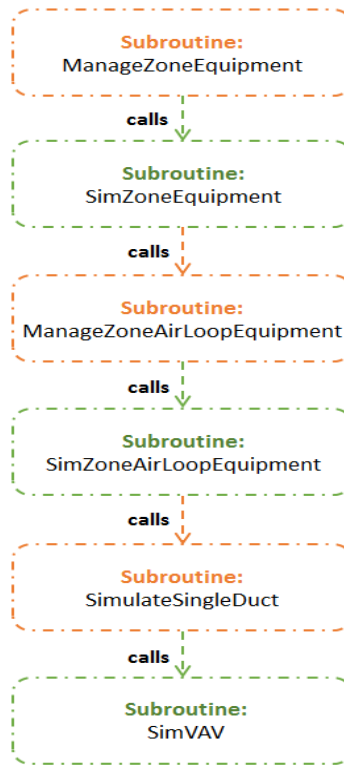


Figure 20 Zone Equipment System Simulation Calling Tree

In order to pass *AirTerminal* back up into “*SimZoneEquipment*”, several parameters should be added in the subroutine in between. Those parameters only function as a carrier of the terminal air flow rate value, and code added into those subroutines are only definitions for the parameter. Notice that for one subroutine, if the

arguments are at the same position in argument list, they represent the same value even if they have different names<sup>3</sup>.

To show exactly how to calculate the total air flow rate for each AHU in loop 1, simplified original and updated EnergyPlus source code are shown as below, in an inverted order from subroutine “*SimVAV*” to subroutine *SimZoneEquipment*”. All subroutine called in loop 1 are renamed as “First-”. All the new added code will be bold.

Some rules of simplified code are denoted below:

1. All the simplified code of subroutines below only contains the parts that related to certain uncertainty. Other parts of the code will be ignored or summarized with a “!” at the beginning and colored with grey.
2. The italic green words started with “!” that stated right after some of the codes will be an explanation for certain codes they followed by.
3. The original arguments of each subroutine will not be shown unless they are part of the calculation for uncertainty in the updated subroutine. Instead, they will just be represented by “OAG”, which is short for “original arguments”.

Original :

SUBROUTINE SimVAV(SysNum,OAG)

1. ! Calculate “*MassFlow*” based on zone load and damper mode.  
SysOutlet(SysNum)%AirMassFlowRate = MassFlow<sup>4</sup>  
Call UpdateSys(SysNum)
  2. ! check if reheat is needed
- 

<sup>3</sup> For instance, subroutine B is defined as “subroutine B(a,b)”. if subroutine A called subroutine B(c,d), it means that a, b are the parameter names in subroutine B, and in subroutine A, subroutine B uses c and d as simulation values for parameter a and b

<sup>4</sup> SysOutlet is an array, with SysNum as its index and AirMassFlowRate is one of the parameter in the array

```

a. ! if reheat is needed

    Then

        i. !recalculate MassFlow based on max reheat air temperature
        ii. SysOutlet(SysNum)%AirMassFlowRate = MassFlow
        iii. ! do the heating coil calculation for each heating coil type

b. Else

    ! do heating coil calculation if reheat is not needed.

End if

END SUBROUTINE SimVAV

```

Updated:

```

SUBROUTINE FirstSimVAV(SysNum,OAG, AirTerminal, ReheatWaterTerminal5)
REAL(r64), INTENT(OUT) :: AirTerminal !store output of air mass flow rate
REAL(r64), INTENT(OUT) :: ReheatWaterTerminal6 !store output of subroutine
ControlCompOutPut

```

```

1. ! Calculate "MassFlow" based on zone load and damper mode.

    SysOutlet(SysNum)%AirMassFlowRate = MassFlow

    AirTerminal = SysOutlet(SysNum)%AirMassFlowRate

    Call UpdateSys(SysNum)

2. ! check if reheat is needed

    a. ! if reheat is needed

        Then

            i. !recalculate MassFlow based on max reheat air temperature
            ii. SysOutlet(SysNum)%AirMassFlowRate = MassFlow

                AirTerminal = SysOutlet(SysNum)%AirMassFlowRate

```

---

<sup>5</sup> for reheat coil hot water distribution

<sup>6</sup> for reheat coil hot water distribution

```

        iii. ! do the heating coil calculation for each heating coil type
    b. Else
        ! do heating coil calculation if reheat is not needed.
    End if
END SUBROUTINE FirstSimVAV

```

The subroutine that called “SimVAV” is “SimulateSingleduct”.

Call of subroutine “AirVAVDistributionUQ” will be added in the updated subroutine to generate uncertainty factors for each terminal only on the first time when this subroutine is under simulation.

Original:

```

SUBROUTINE SimulateSingleDuct(CompName,OAG)
! Find the correct SysNum with the Component Name
SELECT CASE(Sys(SysNum)%SysType_Num)
CASE (SingleDuctVAVReheat)
    Call SimVAV(SysNum, OAG)
END SUBROUTINE SimulateSingleDuct

```

Updated:

```

SUBROUTINE
FirstSimulateSingleDuct(CompName, OAG, AirTerminal, ReheatWaterTerminal 7 ,
AHUNUM)
REAL(r64), INTENT(OUT) :: AirTerminal !store output of subroutine FirstSimVAV

```

---

<sup>7</sup> for reheat coil hot water distribution



```

REAL(r64), INTENT(OUT) :: ReheatWaterTerminal8 !store output of subroutine FirstSimVAV

INTEGER, INTENT(OUT) :: AHUNUM !store output of subroutine FirstSimVAV

CALL AirVAVDistributionUQ

! Find the correct SysNumber with the Component Name

AHUNUM = Sys(SysNum)%AHUNUM

SELECT CASE(Sys(SysNum)%SysType_Num)

CASE (SingleDuctVAVReheat)

    Call FirstSimVAV(SysNum, OAG, AirTerminal, ReheatWaterTerminal)

END SUBROUTINE FirstSimulateSingleDuct

```

The subroutine that called “SimulateSingleDuct” is “*SimZoneAirLoopEquipment*”.

Original:

```

SUBROUTINE SimZoneAirLoopEquipment(AirDistUnitNum, OAG)
DO AirDistCompNum = 1, AirDistUnit(AirDistUnitNum)%NumComponents
  1. SELECT CASE ! select AirDistUnit type
    CASE (SingleDuctVAVReheat)
      CALL
      SimulateSingleDuct(AirDistUnit(AirDistUnitNum)%EquipName(AirDistCompNum)
        ,OAG)
    END SELECT
  2. Do leak mass flow calculation and update the air mass flow if the upstream and
    downstream pipe has air leakage.
END DO
END SUBROUTINE SimZoneAirLoopEquipment

```

---

<sup>8</sup> for reheat coil hot water distribution

Updated:

```
SUBROUTINE FirstSimZoneAirLoopEquipment(AirDistUnitNum, OAG, AirSingleDuct,  
ReheatWaterSingleDuct9,AHUNUM)  
REAL(r64), INTENT(OUT) :: AirSingleDuct !store output of subroutine  
FirstSimulateSingleDuct  
REAL(r64), INTENT(OUT) :: ReheatWaterSingleDuct10 !store output of subroutine  
FirstSimulateSingleDuct  
INTEGER, INTENT(OUT) :: AHUNUM  
DO AirDistCompNum = 1, AirDistUnit(AirDistUnitNum)%NumComponents  
  1. SELECT CASE ! select AirDistUnit type  
    CASE (SingleDuctVAVReheat)  
      CALL FirstSimulateSingleDuct(OAG, AirSingleDuct,  
      ReheatWaterSingleDuct11,AHUNUM)  
    END SELECT  
  2. Do leak mass flow calculation and update the air mass flow if the upstream and  
    downstream pipe has air leakage.  
END DO  
END SUBROUTINE FirstSimZoneAirLoopEquipment
```

The subroutine that called “SimZoneAirLoopEquipment” is  
“ManageZoneAirLoopEquipment”.

Original:

```
SUBROUTINE ManageZoneAirLoopEquipment (OAG)
```

---

<sup>9</sup> for reheat coil hot water distribution

<sup>10</sup> for reheat coil hot water distribution

<sup>11</sup> for reheat coil hot water distribution

```

CALL GetZoneAirLoopEquipment
CALL InitZoneAirLoopEquipment
CALL SimZoneAirLoopEquipment (AirDistUnitNum, OAG)
END SUBROUTINE ManageZoneAirLoopEquipment

```

Updated:

```

SUBROUTINE FirstManageZoneAirLoopEquipment(OAG)
REAL(r64), INTENT(OUT) :: AirEquipSim !store output of subroutine
FirstSimZoneAirLoopEquipment
REAL(r64), INTENT(OUT) :: ReheatWaterEquipSim12 !store output of subroutine
FirstSimZoneAirLoopEquipment
INTEGER, INTENT(OUT) :: AHUNUM
CALL GetZoneAirLoopEquipment
CALL InitZoneAirLoopEquipment
CALL FirstSimZoneAirLoopEquipment(AirDistUnitNum, OAG,AirEquipSim,
ReheatWaterEquipSim13, AHUNUM)
END SUBROUTINE FirstManageZoneAirLoopEquipment

```

The final subroutine that called “ManageZoneAirLoopEquipment” is “SimZoneEquipment”. For this subroutine we didn’t rename it since it’s not any part of the two loops.

Original:

```

SUBROUTINE SimZoneEquipment(FirstHVACIteration, SimAir)

```

1. loop over all supply air path, for each one, loop over and simulate every
- 

<sup>12</sup> for reheat coil hot water distribution

<sup>13</sup> for reheat coil hot water distribution

component in one path (components contain zone splitter and supply plenum)

```
2. DO ControlledZoneNum = 1, NumOfZones
    a. DO EquipTypeNum = 1,
        ZoneEquipList(ControlledZoneNum)%NumOfEquipTypes
            SELECT CASE
                (PrioritySimOrder(EquipTypeNum)%EquipType_Num)
                CASE(AirDistUnit_Num) ! 'ZoneHVAC:AirDistributionUnit'
                    CALL ManageZoneAirLoopEquipment (OAG)
            END SELECT
            CALL UpdateSystemOutputRequired
        END DO
    END DO
3. loop over all supply air path again
4. simulate return air path and balance the air mass flow rate in each loop.
END SUBROUTINE SimZoneEquipment
```

As shown in the code above, this subroutine will loop over all the zones (step 2), and inside the loop it'll loop over all equipment that belongs to the certain zone with an internal do-loops. Thus, for loop 1 in updated subroutine (new step 2), total air mass flow rate will be calculated on equipment level first at the end of each internal loop for each AHU separately.

To determine which air mass flow rate of zone component (carried by “*AirEquip*” in updated “*SimZoneEquipment*”) should be added up together, we have to first know the AHU number that each zone equipment belongs to. To make the process clear and easy, we add a new column in the uncertainty module in IDF file for users to classify the zones based on their related AHU. In EnergyPlus source code, we add following parameters

into an array “Sys” which carries parameters for all AHU systems. This array is a part of the original source code of EnergyPlus.

```
REAL(r64)  :: AirFlowUQRo = 0.0D0  
REAL(r64)  :: AirFlowUQXi = 0.0D0  
INTEGER    :: AHUNUM  
REAL(r64)  :: UQRoReheat  
REAL(r64)  :: UQXiReheat
```

“AHUNUM” is the AHU number linked with each zone component that set by users. The other four new parameters are uncertainty factors for each AHU. Values of all these five parameters will be either read from the IDF file or generated during the simulation of the uncertainty module, and then be stored in the array “Sys”.

To locate system parameters for each AHU, we keep using the parameter called “Sysnum”, which is the one that original EnergyPlus source code uses. “Sysnum” is the index number for AHU systems in this array. Under each “Sysnum” there’s a group of all other parameters for each AHU system, including its AHUNUM and uncertainty factors. When simulating components for each zone, each component can find its own “Sysnum” with its component name or sys name, so that certain TotalAirAHU that each AirTerminal should be added into can be located.

A second do-loop (loop 2) for air mass flow rate simulation will be added after loop1. Loop 2 will assign total air mass flow rate to each air terminal. All subroutine called in loop 2 are renamed as “Re-”. Also by using “Sysnum” in loop2, the unique uncertainty factors for each component can be found.

Both new added loop 1 and loop 2 are similar with the old do-loops, but with some new added parameters and equations.

Updated:

```
SUBROUTINE SimZoneEquipment(FirstHVACIteration, SimAir)
```

```
!Define new added parameters:
```

```
REAL(r64), ALLOCATABLE, DIMENSION(:) :: TotalAirAHU ! Total air supplied by each AHU
```

```
REAL(r64) :: TotalReheatWaterZone=0.0d014 ! Total VAV water flowrate total for each the zone
```

```
REAL(r64) :: TotalReheatWater=0.0d015 ! Sum total reheat water flow from all zones
```

```
REAL(r64) :: AirEquip = 0.0d0 ! Air supplied to the zone by each equipment
```

```
REAL(r64) :: ReheatWaterEquip = 0.0d016 ! Reheat Water to the zone by each equipment
```

```
INTEGER :: AHUNUM
```

```
CHARACTER (len=MaxNameLength) :: CurrentModuleObject !for ease in getting objects
```

```
INTEGER :: CountAHU
```

```
! read CountAHU from userinput in newly added uncertainty module
```

```
"SystemUQ:AirDistributionNetwork:VAVReheat"
```

```
CurrentModuleObject = 'SystemUQ:AirDistributionNetwork:VAVReheat'
```

```
CountAHU = GetNumObjectsFound(TRIM(CurrentModuleObject))
```

```
Allocate (TotalAirAHU(CountAHU))
```

```
TotalAirAHU = 0.0d0
```

```
1. ! loop over all supply air path, for each one, loop over and simulate every
```

---

<sup>14</sup> for reheat coil hot water distribution

<sup>15</sup> for reheat coil hot water distribution

<sup>16</sup> for reheat coil hot water distribution

component in one path (components contain zone splitter and supply plenum)

2. DO ControlledZoneNum = 1, NumOfZones
  - a. DO EquipTypeNum = 1, ZoneEquipList(ControlledZoneNum)%NumOfEquipTypes

**AirEquip = 0.0d0**

**ReheatWaterEquip = 0.0d0<sup>17</sup>**

SELECT CASE !select EquipType

CASE(AirDistUnit\_Num) ! 'ZoneHVAC:AirDistributionUnit'

CALL **FirstManageZoneAirLoopEquipment** (OAG, **AirEquip**,  
**ReheatWaterEquip**, **AHUNUM**)

**TotalAirAHU(AHUNUM) = TotalAirAHU(AHUNUM) + AirEquip**

**TotalReheatWaterZone<sup>18</sup> = TotalReheatWaterZone +**

**ReheatWaterEquip**

END SELECT

END DO

**TotalReheatWater<sup>19</sup> = TotalReheatWater + TotalReheatWaterZone**

END DO

3. DO ControlledZoneNum = 1, NumOfZones
  - a. DO EquipTypeNum = 1, ZoneEquipList(ControlledZoneNum)%NumOfEquipTypes

SELECT CASE !select EquipType

CASE(AirDistUnit\_Num) ! 'ZoneHVAC:AirDistributionUnit'

CALL **ReManageZoneAirLoopEquipment** (OAG, **TotalAirAHU**,  
**TotalReheatWater<sup>20</sup>**, **CountAHU**)

END SELECT

CALL *UpdateSystemOutputRequired*

---

<sup>17</sup> for reheat coil hot water distribution

<sup>18</sup> for reheat coil hot water distribution

<sup>19</sup> for reheat coil hot water distribution

<sup>20</sup> for reheat coil hot water distribution

```

        END DO
    END DO
4. ! loop over all supply air path again
5. ! simulate return air path and balance the air mass flow rate in each loop.
END SUBROUTINE SimZoneEquipment

```

As shown above, the subroutine *“UpdateSystemOutputRequired”* will not be called until the end of loop2. This will make sure that the zone demands will not be changed before loop 2 is taking place, so that the second loop will be simulated under the same condition as loop1, which is also the same as the original situation.

The updates for each subroutine called in loop 2 are basically the same with those in loop 1. The equations that capture the uncertainty impact will be added in *“ReSimVAV”*, and new parameters will be added in between subroutines to carry and pass the value. The subroutine called in loop2 is *“ReManageZoneAirLoopEquipment”*.

```

Updated:

SUBROUTINE
ReManageZoneAirLoopEquipment(OAG,TotalAirAHU,TotalReheatWater21,TotalAHUNUM
M)
CALL ReSimZoneAirLoopEquipment(AirDistUnitNum, OAG, TotalAirAHU,
TotalReheatWater22,TotalAHUNUM)
END SUBROUTINE ReManageZoneAirLoopEquipment

```

The subroutine called next is *“ReSimZoneAirLoopEquipment”*

---

<sup>21</sup> for reheat coil hot water distribution

<sup>22</sup> for reheat coil hot water distribution



Updated:

```
SUBROUTINE ReSimZoneAirLoopEquipment(AirDistUnitNum, OAG, TotalAir,  
TotalReheatWater23,TotalAHUNUM)  
DO AirDistCompNum = 1, AirDistUnit(AirDistUnitNum)%NumComponents  
  1. SELECT CASE !select EquipType  
    CASE (SingleDuctVAVReheat)  
    CALL  
      ReSimulateSingleDuct(AirDistUnit(AirDistUnitNum)%EquipName(AirDistC  
ompNum),OAG, TotalAir, TotalAHUNUM, TotalReheatWater24)  
    END SELECT  
  2. ! Do leak mass flow calculation and update the air mass flow if the upstream  
    and downstream pipe has air leakage.  
END DO  
END SUBROUTINE ReSimZoneAirLoopEquipment
```

The subroutine called next is “*ReSimulateSingleDuct*”.

Updated:

```
SUBROUTINE ReSimulateSingleDuct(CompName, OAG, TotalAir, TotalAHUNUM,  
TotalReheatWater25)  
! Find the correct SysNumber with the Component Name  
TotalAirAHU = TotalAir(Sys(SysNum)%AHUNUM)  
SELECT CASE(Sys(SysNum)%SysType_Num)
```

---

<sup>23</sup> for reheat coil hot water distribution

<sup>24</sup> for reheat coil hot water distribution

<sup>25</sup> for reheat coil hot water distribution

CASE (SingleDuctVAVReheat) ! SINGLE DUCT:VAV:REHEAT

Call **ReSimVAV**(SysNum, OAG, **TotalAirAHU**, **TotalReheatWater**<sup>26</sup>)

END SUBROUTINE ReSimulateSingleDuct

The subroutine called at last is “ReSimVAV”.

Updated:

SUBROUTINE ReSimVAV(SysNum, OAG, **TotalAirAHU**, **TotalReheatWater**<sup>27</sup>)

**REAL(r64), INTENT(IN) :: TotalAirAHU** *!get TotalAir for next level subroutine*

**REAL(r64), INTENT(IN) :: TotalReheatWater**<sup>28</sup> *!get TotalReheatWater for next level subroutine*

1. ! Calculate “MassFlow” based on zone load and damper mode.

SysOutlet(SysNum)%AirMassFlowRate = MassFlow

Call UpdateSys(SysNum)

2. QActualHeating = QToHeatSetPt - Massflow \* CpAirZn \*

(SysInlet(SysNum)%AirTemp-ZoneTemp)

*! check if reheat is needed*

- a. If((MassFlow > SmallMassFlow .OR. Sys(SysNum)%DamperHeatingAction .EQ.  
ReverseAction) .AND. &

(QActualHeating > 0.0) .AND. (TempControlType(ZoneNum) .NE.

SingleCoolingSetPoint) ) *! if reheat is needed*

Then

- i. !recalculate MassFlow based on max reheat air temperature

---

<sup>26</sup> for reheat coil hot water distribution

<sup>27</sup> for reheat coil hot water distribution

<sup>28</sup> for reheat coil hot water distribution

```

SysOutlet(SysNum)%AirMassFlowRate = (MassFlow +
Sys(SysNum)%AirFlowUQXi*TotalAirAHU)*(1.0d0 +
Sys(SysNum)%AirFlowUQRo)
MassFlow = SysOutlet(SysNum)%AirMassFlowRate

```

```

ii. ! do the heating coil calculation for each heating coil type

```

```

b. Else

```

```

SysOutlet(SysNum)%AirMassFlowRate = (MassFlow +
Sys(SysNum)%AirFlowUQXi*TotalAirAHU)*(1.0d0 +
Sys(SysNum)%AirFlowUQRo)
MassFlow = SysOutlet(SysNum)%AirMassFlowRate

```

```

! do heating coil calculation if reheat is not needed.

```

```

End if

```

```

END SUBROUTINE ReSimVAV

```

The updated Working Process of Subroutine: ManageZoneEquipment and zone equipment simulation calling tree is shown as below.

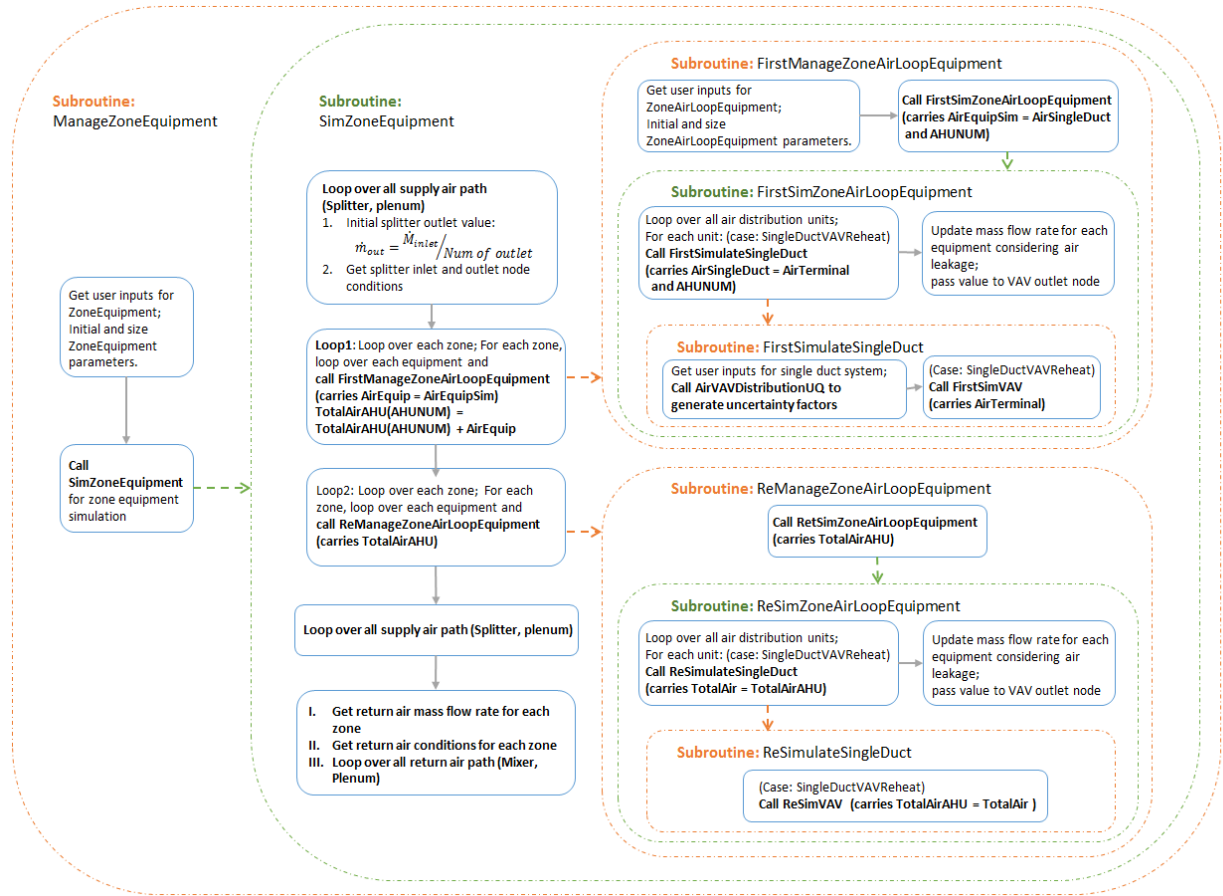


Figure 21 updated Working Process of Subroutine: ManageZoneEquipment

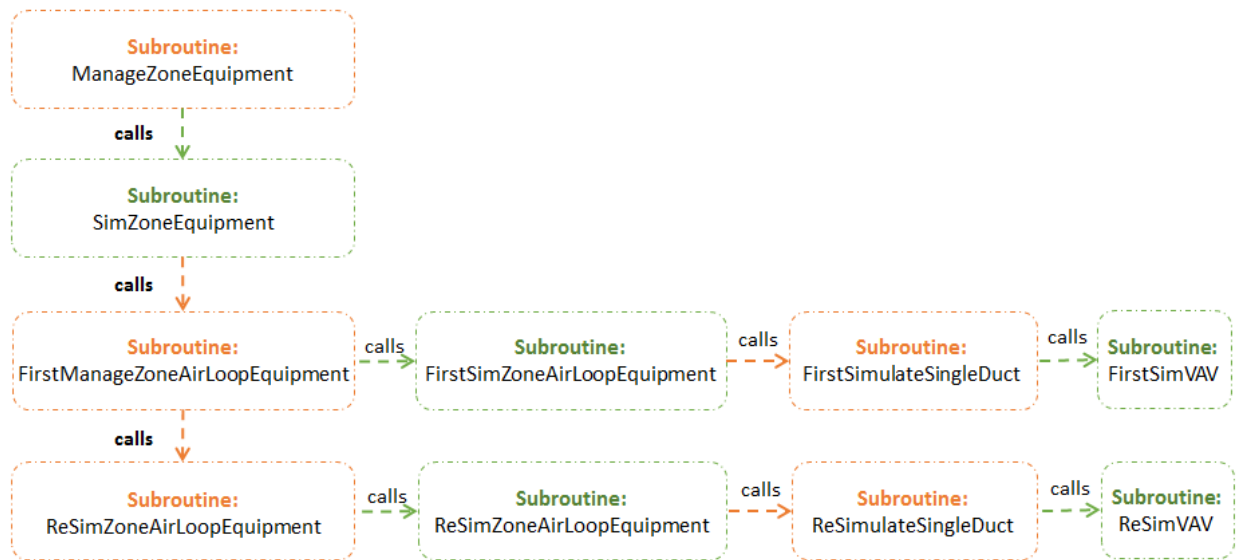


Figure 22 Updated Zone Equipment System Simulation Calling Tree

## Implements in IDF file

A new module is added into IDF file for user to enter the 2 uncertainty factors and assign zone equipment to AHU. A screenshot of the module is shown as below.

Class List

Uncertainty Quantification: HVAC System

- [0001] SystemUQ:SystemUQControl
  - [.....] SystemUQ:ChillerElectricEIR
  - [.....] SystemUQ:VariableSpeedPump
  - [.....] SystemUQ:VariableVolumeFan
  - [.....] SystemUQ:VariableSpeedCoolingTower
  - [.....] SystemUQ:HeatingCoilUA
  - [.....] SystemUQ:TemperatureController
  - [.....] SystemUQ:OACController
- [0001] SystemUQ:ChilledWaterDistributionNetwork
- [0001] SystemUQ:HotWaterDistributionNetwork
- [0004] SystemUQ:AirDistributionNetwork:VAVReheat

Comments from IDF

Explanation of Keyword

Required total air flowrate deviates from delivered  
The network unbalance uncertainty

ID: N2  
Default: 0.1]

Field	Units	Obj1	Obj2	Obj3	Obj4
AirLoop AHU Name		VAV_1	VAV_2	VAV_3	VAV_5
AHU Fan Massflow Uncertain Factor		0.1	0.1	0.1	0.1
Air Distribution Network Uncertain Factor		0.1	0.1	0.1	0.1
Component 1 Name		Core_bottom VAV B	Core_mid VAV Box I	Core_top VAV Box I	Basement VAV Box
Component 2 Name		Perimeter_bot_ZN_	Perimeter_mid_ZN_	Perimeter_top_ZN_	
Component 3 Name		Perimeter_bot_ZN_	Perimeter_mid_ZN_	Perimeter_top_ZN_	
Component 4 Name		Perimeter_bot_ZN_	Perimeter_mid_ZN_	Perimeter_top_ZN_	
Component 5 Name		Perimeter_bot_ZN_	Perimeter_mid_ZN_	Perimeter_top_ZN_	
Component 6 Name					

Figure 23 Added Module in IDF File for UQ: Air Distribution

## System UQ: Water Distribution in VAV Reheat Coils

The causes and quantifications for uncertainties in VAV reheat coil hot water distribution are the same with those for water distribution in AHU coils (2.d).

$$\dot{m}_i^r = (\dot{m}_i^d + x_i \dot{M}^d)(1 + \rho)$$

$\dot{m}_i^r$ : Actual water mass flow rate received by terminal I

$\dot{m}_i^d$ : Demand water mass flow rate by terminal I

$\dot{M}^d$ : Total demand water mass flow rate by one pump

$\rho$ : Uncertainty factor that capture the error in the use of idealized pump reaction

$x_i$ : Uncertainty factor caused by unbalance of distribution

### Implement in EnergyPlus:

The reheat coil simulation is also in subroutine “SimVAV”, right after air mass flow rate is being calculated.

Following picture is the same with Figure 19 Working Process of Subroutine: SimVAV, with simulation for reheat coils marked in red.

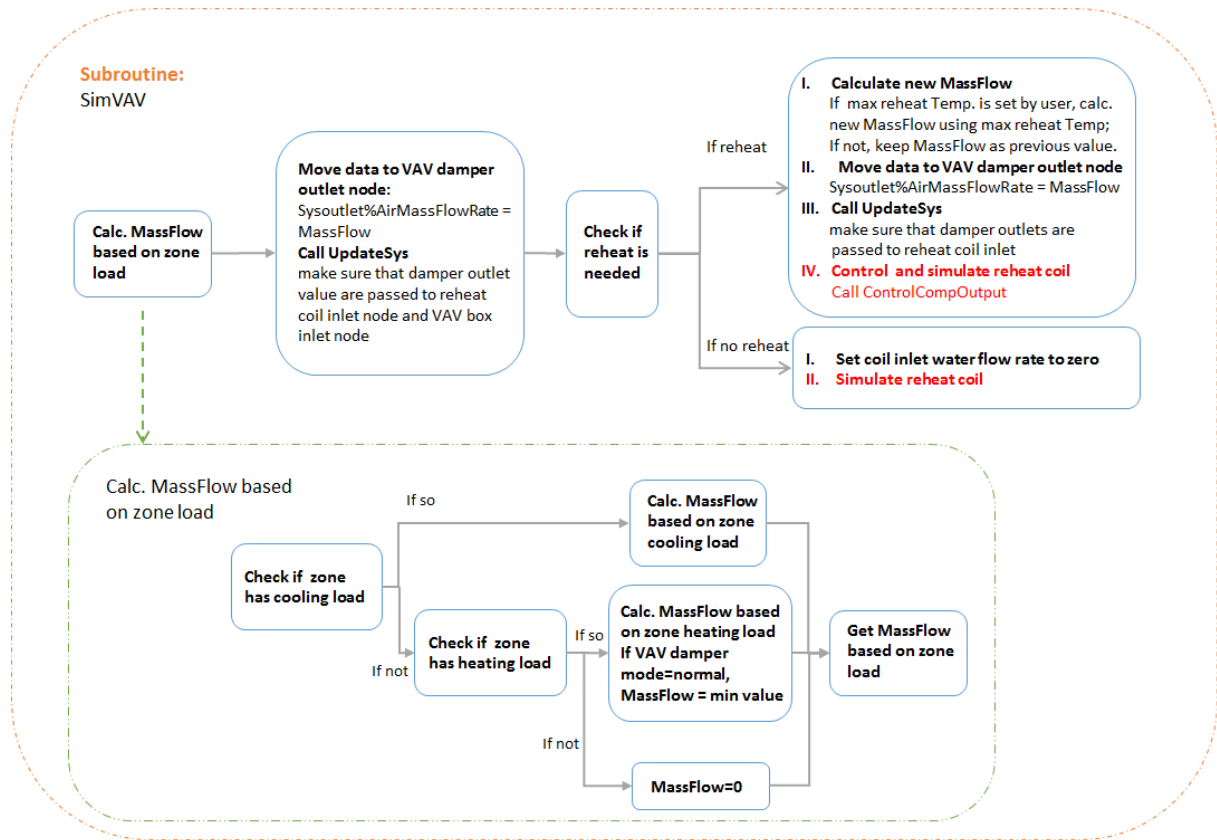


Figure 24 Working Process of Subroutine SimVAV with Simulation of Reheat Coils Marked in Red

Thus the implementation of reheat water distribution uncertainty will be also in those subroutines introduced in the previous section, following the same steps.

1. Generate and assign  $UQRoReheat(\rho)$  and  $UQXiReheat(x_i)$ , to each zone equipment

For factor UQRoReheat, the implementation for reheat coils and AHU heating coils are sharing the same  $\rho$  factor since they are all connected to one pump. This factor will be generated during AHU coil simulation. During zone equipment simulation, this factor will be allocated in subroutine *"AirVAVDistributionUQ"*, along with the generation of air distribution uncertainty factors.

For factor UQXiReheat , it will also be generated in subroutine *"AirVAVDistributionUQ"*.

## 2. Calculate total actuated reheat water flow rate $\dot{M}^d$

Same as air distribution, terminal reheat water flow rate, carried by *"ReheatWaterTerminal"*, will be calculated and extracted from subroutine *"FirstSimVAV"*, and passed through subroutine *"FirstSimulateSingleDuct"*, *"FirstSimZoneAirLoopEquipment"*, *"FirstManageZoneAirLoopEquipment"*, and eventually into the updated subroutine *"SimZoneEquipment"* to be added up to get *"TotalReheatWater"* at the end of loop1. Implementation in subroutines between are all shown in the previous section in simplified EnergyPlus source code. Only two difference are in subroutine *"FirstSimVAV"* and updated *"SimZoneEquipment"*.

One difference in *"SimZoneEquipment"* is that since all the reheat coils are connected to one pump, the total reheat water will be the sum of all terminal reheat water flow rates instead of sum for each AHU separately. The parameter added to carry this value is *"TotalReheatWater"*

Since subroutine *"SimZoneEquipment"* loops over all the zones and for each zone, it loops over all the equipment with an internal do-loops, total reheat water mass flow rate will be calculated on equipment level first at the end of each internal loop, value carried by *"TotalReheatWaterZone"*, then on the zone level at the end of external loop, value carried by *"TotalReheatWater"*.

The updated EnergyPlus source code is shown in the previous section.

The other difference is in “FirstSimVAV”. The calculation of reheat water flow rate is similar with that for AHU coils, only with a built-in water coil controller instead of the user added controller module. This built-in controller is called “ControlCompOutput”. The working process of this controller is illustrated as below.

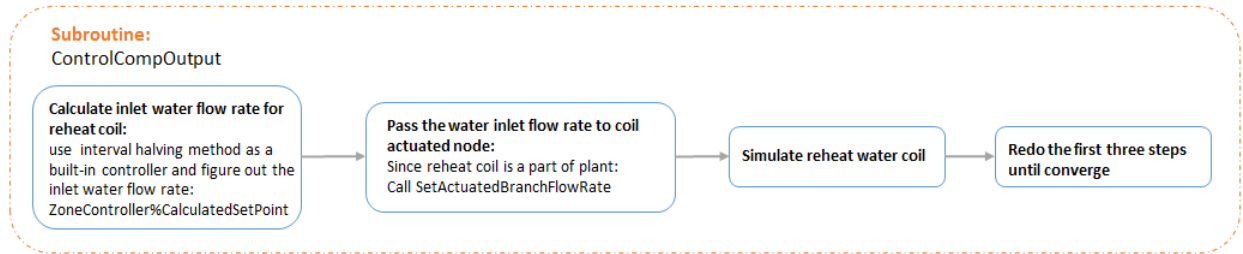


Figure 25 Working Process of Subroutine: ControlCompOutput

The subroutine “ControlCompOutput” first uses interval halving method to find out the proper reheat water mass flow rate for each VAV component, then call subroutine “SetActuatedBranchFlowRate” to pass this value to actuated node for simulating reheat coil component. Same as for AHU coils, in the first loop, actuated value for reheat water should be extract before it got pass into other node, which is before calling the subroutine “SetActuatedBranchFlowRate”. We store this value in a new parameter “ReheatControllerActuatedValue”

Simplified original and updated EnergyPlus source code of “SimVAV” and “ControlCompOutput” are shown as below.

Updated ReSimVAV:

```
SUBROUTINE ReSimVAV(SysNum, OAG, ReheatWaterTerminal)
```

```
!Calculate and update air mass flow rate for VAV terminal.
```

```
! IF reheat
```

```
    SELECT CASE(Sys(SysNum)%ReheatComp_Num)
```



```

CASE(HCoilType_SimpleHeating) ! COIL:WATER:SIMPLEHEATING
CALL ReControlCompOutput(OAG, TotalReheatWaterGeneral =
    TotalReheatWater, &
    UQRoReheatGeneral = Sys(SysNum)%UQRoReheat, &
    UQXiReheatGeneral = Sys(SysNum)%UQXiReheat)
ELSE
SELECT CASE(Sys(SysNum)%ReheatComp_Num)
CASE(HCoilType_SimpleHeating) ! COIL:WATER:SIMPLEHEATING
DummyMdot = 0.d0
CALL SetActuatedBranchFlowRate(DummyMdot,OAG)
!call the reheat coil with the NO FLOW condition to make sure that the Node
values are passed through to the coil outlet correctly
CALL SimulateWaterCoilComponents(Sys(SysNum)%ReheatName,OAG)
ENDIF
END SUBROUTINE FirstSimVAV

```

Original FirstControlCompOutput:

```

SUBROUTINE FirstControlCompOutput(OAG)
!Calculated the set point for reheat water mass flow rate
IF (PRESENT(LoopNum)) THEN ! if this is a plant component
    CALL SetActuatedBranchFlowRate(ZoneController%CalculatedSetPoint,OAG) !
    pass values to plant nodes and actuated node
ELSE ! not a plant component
    Node(ActuatedNode)%MassFlowRate = ZoneController%CalculatedSetPoint
ENDIF
SELECT CASE(SimCompNum)
CASE(3) ! 'COIL:HEATING:WATER'
CALL SimulateWaterCoilComponents(OAG) ! Simulate reheat coil for the VAV
system

```

END SELECT

END SUBROUTINE FirstControlCompOutput

Updated FirstControlCompOutput:

SUBROUTINE FirstControlCompOutput(OAG, **ReheatControllerActuatedValue**)

!Calculated the set point for reheat water mass flow rate

**ReheatControllerActuatedValue = ZoneController%CalculatedSetPoint**

IF (PRESENT(LoopNum)) THEN *! if this is a plant component*

CALL SetActuatedBranchFlowRate(ZoneController%CalculatedSetPoint,OAG) !

*pass values to plant nodes and actuated node*

ELSE *! not a plant component*

Node(ActuatedNode)%MassFlowRate = ZoneController%CalculatedSetPoint

ENDIF

SELECT CASE(SimCompNum)

CASE(3) *! 'COIL:HEATING:WATER'*

CALL SimulateWaterCoilComponents(OAG) *! Simulate reheat coil for the VAV system*

END SELECT

END SUBROUTINE FirstControlCompOutput

Update  $\dot{m}_i^d$  to  $\dot{m}_i^r$  using  $\dot{m}_i^r = (\dot{m}_i^d + x_i \dot{M}^d)(1 + \rho)$

After “**TotalReheatWater**” is calculated at the end of loop1 in, this value will be passed into “*ReSimVAV*” in loop2 with the same process as for air distribution. The updated subroutines are showed in previous section except for “*ReSimVAV*”.

In loop2, “*ResimVAV*” will call an updated “*ControlCompOutput*”, named as “*ReControlCompOutput*”, which contains the equation of adding uncertainties to the actuated value of reheat water flow rate. The simplified EnergyPlus source code of “*ReSimVAV*” and “*ReControlCompOutput*” are shown as below.

Updated FirstSimVAV:

SUBROUTINE FirstSimVAV(Sysnum, OAG, **TotalReheatWater**)

!Calculate and update air mass flow rate for VAV terminal.

!If reheat

SELECT CASE(Sys(SysNum)%ReheatComp\_Num)

CASE(HCoilType\_SimpleHeating) *! COIL:WATER:SIMPLEHEATING*

CALL FirstControlCompOutput(OAG,

**ReheatControllerActuatedValue=ReheatWaterTerminal)**

ELSE

SELECT CASE(Sys(SysNum)%ReheatComp\_Num)

CASE(HCoilType\_SimpleHeating) *! COIL:WATER:SIMPLEHEATING*

DummyMdot = 0.d0

CALL SetActuatedBranchFlowRate(DummyMdot,OAG)*!call the reheat coil  
with the NO FLOW condition to make sure that the Node values are  
passed through to the coil outlet correctly*

CALL SimulateWaterCoilComponents(OAG)

ENDIF

END SUBROUTINE ReSimVAV

Updated ReControlCompOutput:

SUBROUTINE ReControlCompOutput(OAG,

**TotalReheatWater,UQRoReheat,UQXiReheat)**

!Calculated the set point for reheat water mass flow rate

***ZoneController%CalculatedSetPoint = (1.0d0 + UQRoReheat) ×  
(ZoneController%CalculatedSetPoint + TotalReheatWater ×  
UQXiReheat)***

IF (PRESENT(LoopNum)) THEN *! this is a plant component*

```

        CALL SetActuatedBranchFlowRate(ZoneController%CalculatedSetPoint, OAG)
ELSE ! not a plant component
        Node(ActuatedNode)%MassFlowRate = ZoneController%CalculatedSetPoint
ENDIF
SELECT CASE(SimCompNum)
        CASE(3) ! 'COIL:HEATING:WATER'
                CALL SimulateWaterCoilComponents(OAG) ! Simulate reheat coil for the
                VAV system
END SELECT
END SUBROUTINE ReControlCompOutput

```

Above is the implementation for water distribution uncertainty in VAV reheat coils. Notice that we capture the water distribution uncertainty for AHU coils and VAV reheat coils separately, which means that we consider that the water supplied by pump to AHU coils will not go into any of the VAV reheat coils. The reason why we implemented in this way is because that the method we use to mimic the unbalance of water distribution needs to loop over all the coils twice, thus if we want to calculate the unbalancing of AHU coils and VAV reheat coils together, we have to find the loop that contains both AHU coil simulation and VAV reheat coil simulation, and simulate this loop twice. However, as introduced in chapter 4, the simulations for AHU coils is in loop Manage Air Loop while simulation for VAV reheat coils is in loop Manage Zone Equipment, and the exterior loop that contains this two simulations is the Manage HVAC. To simulate the loop Manage HVAC will be too complicated since it contains all the simulation for building HVAC system. Therefore, we implement the water distribution for this two parts separately.

## **CHAPTER 5**

### **SIMULATION RESULTS ANALYSIS**

To test how much the five developed uncertainties are going to impact the energy consumption, we use GURA-W, the Georgia Tech Uncertainty and Risk Analysis workbench, specifically developed to carry out uncertainty analysis based on Monte Carlo simulations (Lee et.al. 2013). In this study we use it to generate samples for each uncertainty factor within a certain range, and run the simulations for both January (typical heating dominated month) and July (typical cooling dominated month) and get hourly energy consumptions of the building. Units for all considered simulation outcomes results are Joule.

#### **Heating coil UA factor uncertainty**

For the uncertainty of the UA factor, since it only has impact on the heating coils which will only be used under cooling condition, we run the simulation for only January. 50 groups of samples were generated for the uncertainty of each heating coil in the system. All the samples are within the chosen uncertainty range from 0.8 to 1.2.

The simulation result is plotted as below. To make the result easier to read, we calculate the hourly difference between simulation result with uncertainty and those with no uncertainty which are regarded as the baseline.

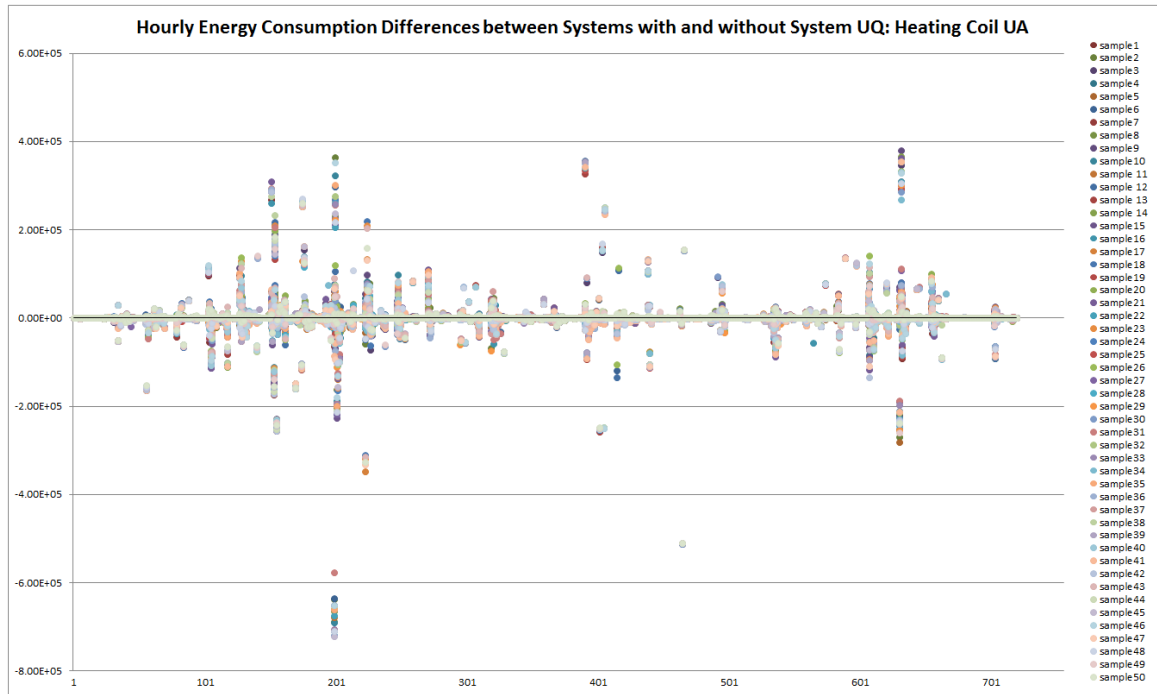


Figure 26 Hourly Energy Consumption Difference between Systems with and without Heating Coil UA Uncertainty

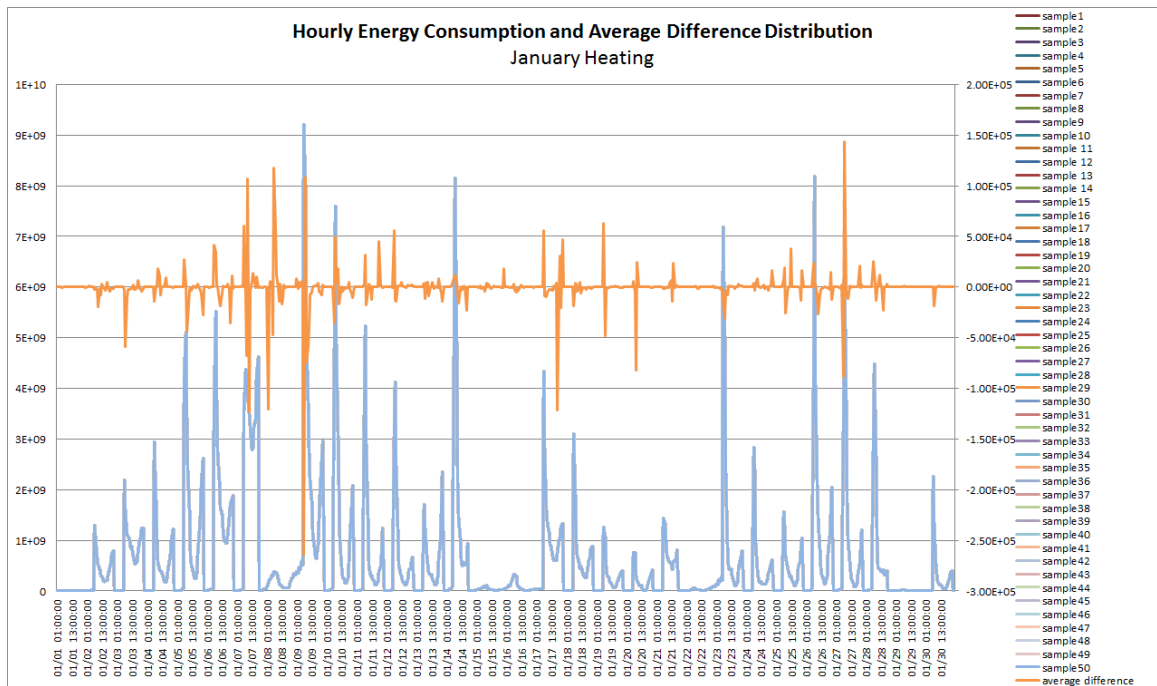


Figure 27 Hourly Energy Consumption and Hourly Average Energy Consumption Difference Distributions for UQ: Heating Coil UA

According to Figure 26, most of the result points are close to 0 except for a few ones. Since there are 36,000 data points in Figure 26, and only less than 10 of them have

significant high values, those outliers will not have significant impacts on the overall average value of the energy difference. Figure 27 shows the hourly energy consumption distributions for all 50 samples (left y-axis) and the average hourly energy consumption difference distribution (right y-axis). The hourly average difference doesn't have any obvious relevance with hourly energy consumption, and the deviations are relatively small compared to the energy consumptions.

We show the maximum and minimum boundary lines for both hourly and daily energy consumption differences, and plot the monthly energy consumption difference over all 50 samples in Figure 28.

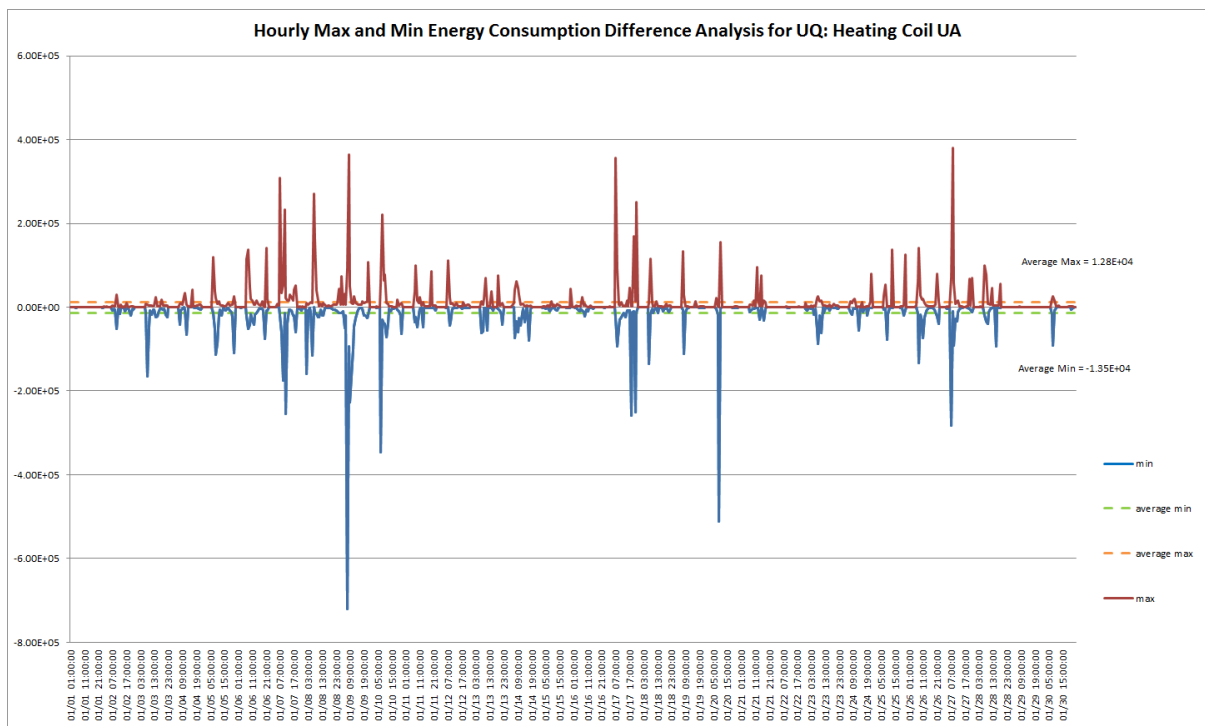


Figure 28 Hourly Max and Min Energy Consumption Difference Analysis for UQ: Heating Coil

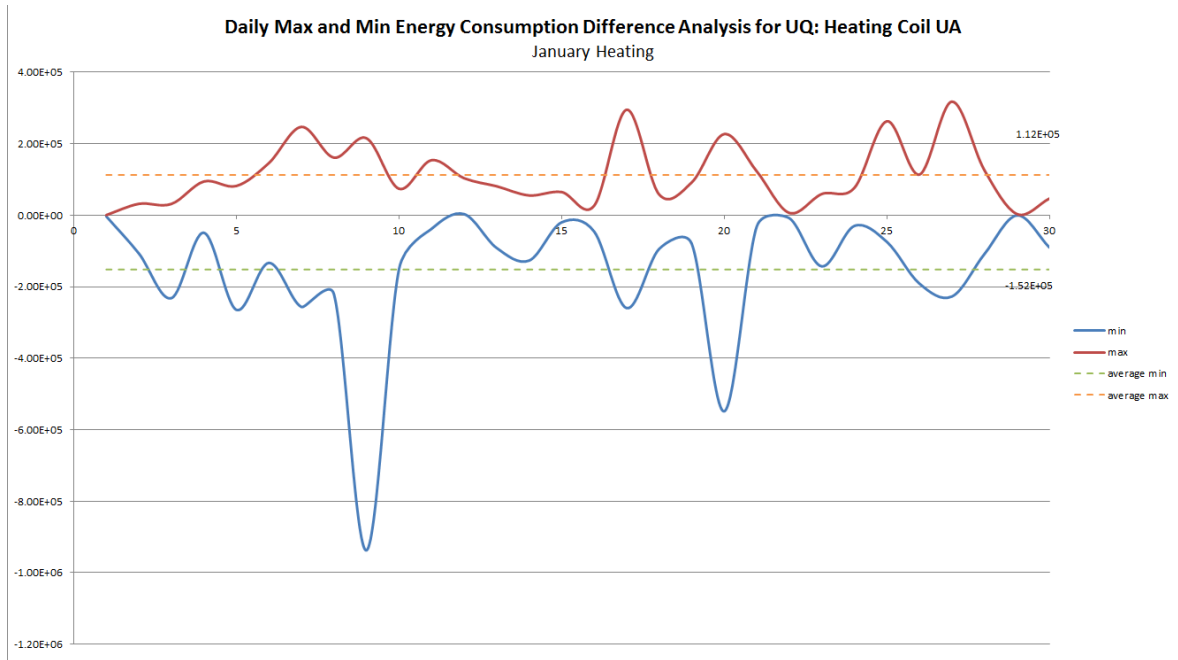


Figure 29 Daily Max and Min Energy Consumption Difference Analysis for UQ: Heating Coil UA

Table 1 Daily Average Energy Consumption with UQ: Heating Coil UA

Date	1	2	3	4	5	6	7	8	9	10
average energy consumption (J)	8.66E+07	8.26E+09	1.64E+10	1.24E+10	2.74E+10	3.31E+10	4.41E+10	4.70E+09	4.85E+10	3.06E+10
Date	11	12	13	14	15	16	17	18	19	20
average energy consumption (J)	1.81E+10	1.51E+10	1.17E+10	2.64E+10	9.22E+08	2.25E+09	2.12E+10	1.27E+10	2.08E+10	1.03E+11
Date	21	22	23	24	25	26	27	28	29	30
average energy consumption (J)	9.17E+09	7.28E+08	2.11E+10	9.83E+09	8.70E+09	2.79E+10	2.11E+10	1.43E+10	9.24E+07	6.40E+09

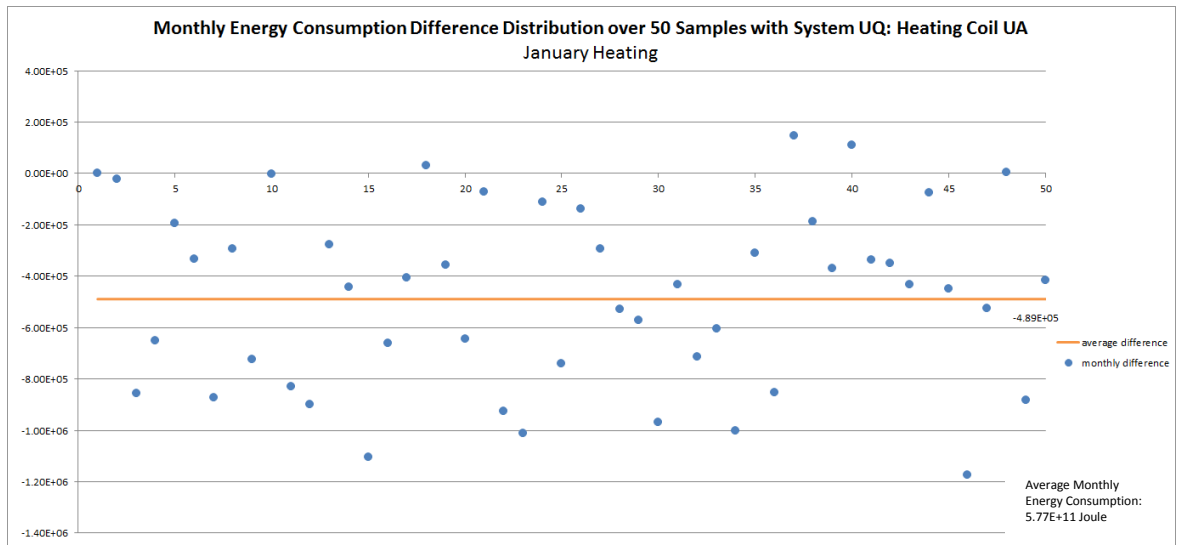


Figure 30 Monthly Energy Consumption Difference Distribution over 50 Samples for System UQ: Heating Coil UA



Figure 28 and Figure 29 shows the average range of hourly and daily energy consumption difference that a  $\pm 20\%$  uncertain in heating coil UA factor can cause: hourly difference range from  $-1.35\text{E}+4$  ( $-0.002\%$  compare to baseline) to  $1.28\text{E}+4$  ( $0.002\%$  compare to baseline), and daily difference range from  $-1.52\text{E}+4$  ( $-0.001\%$  compare to baseline) to  $1.12\text{E}+4$  ( $0.001\%$  compare to baseline). The results also shows that the uncertainty can do both good and bad to the value of energy consumption quite equally since we set the range of uncertainty to  $\pm 20\%$ . Figure 30 shows a distribution of monthly extra heating over all the samples, with an average value of  $-4.89\text{E}+5$  (less than  $0.001\%$  compare to baseline value). It shows that hourly positive energy consumption difference doesn't offset the negative difference. However, although it is normal for heating coils to not act exactly with their rated value, this less than  $\pm 0.01\%$  of energy consumption difference seems to be inevitable but also trivial. Of course a deeper inspection in future studies would have to confirm that the  $\pm 20\%$  range is an accurate assumption.

### **Temperature sensor uncertainty**

This also leads to a very small difference. Since the hourly difference are too small to be shown, we ran the daily simulation instead for two months for both cooling days and heating days, with a uncertainty range from 0.8 to 1.2 as well. The daily results are still very small.

The reason for this surprisingly small result may be that since we only change the temperature sensor at the outlet of heating and cooling coils which only affect the temperature of supply air that goes into the zone, but not the sensor in the zone that will pass the zone temperature back to the HVAC systems. Although the supply air temperature will not reach its setpoints precisely, as long as the zone is giving back the

correct zone temperature, the HVAC systems will keep working until the zone temperature feedback reaches its setpoint. In this case, the energy consumptions of system with and without the temperature sensor uncertainty will not differ much. We can thus conclude that the sensed supply air temperature has no effect on energy consumption because of the zone will ask for energy until its need is satisfied, irrespective of the sensor reading on outlet temperature. To further investigate the impact of temperature sensor uncertainty, it's better to add the uncertainty on zone temperature sensors.

### OA controller uncertainty

We run simulations and get hourly energy consumption for this uncertainty for both January and July, each season has 50 groups of samples, all range from 0.8 to 1.2.

The simulation results are plotted as below.

#### a. January

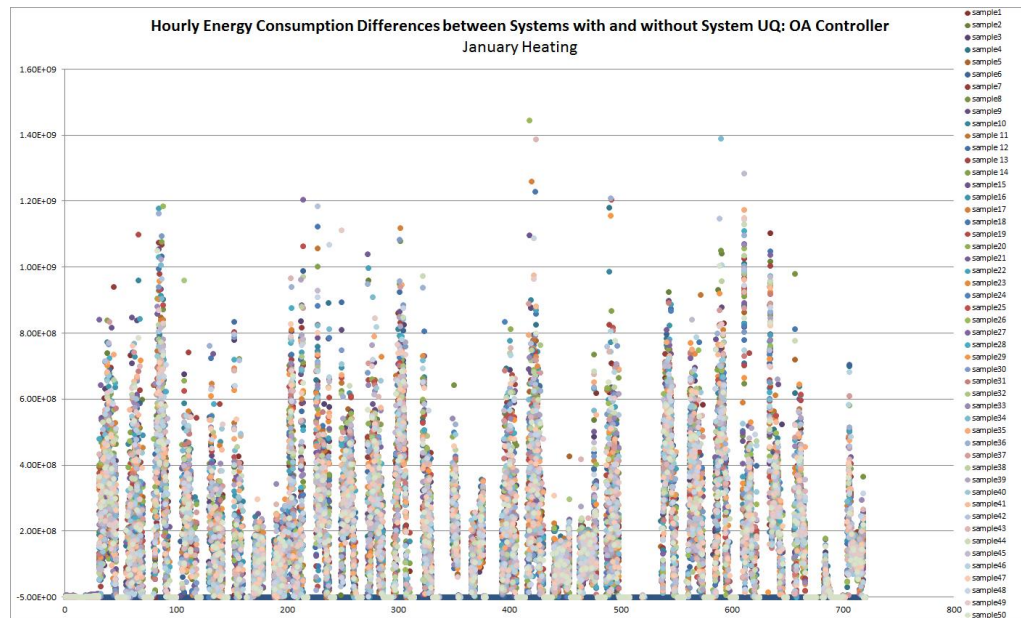


Figure 31 Hourly Energy Consumption Differences between Systems with and without System UQ: OA Controller, January

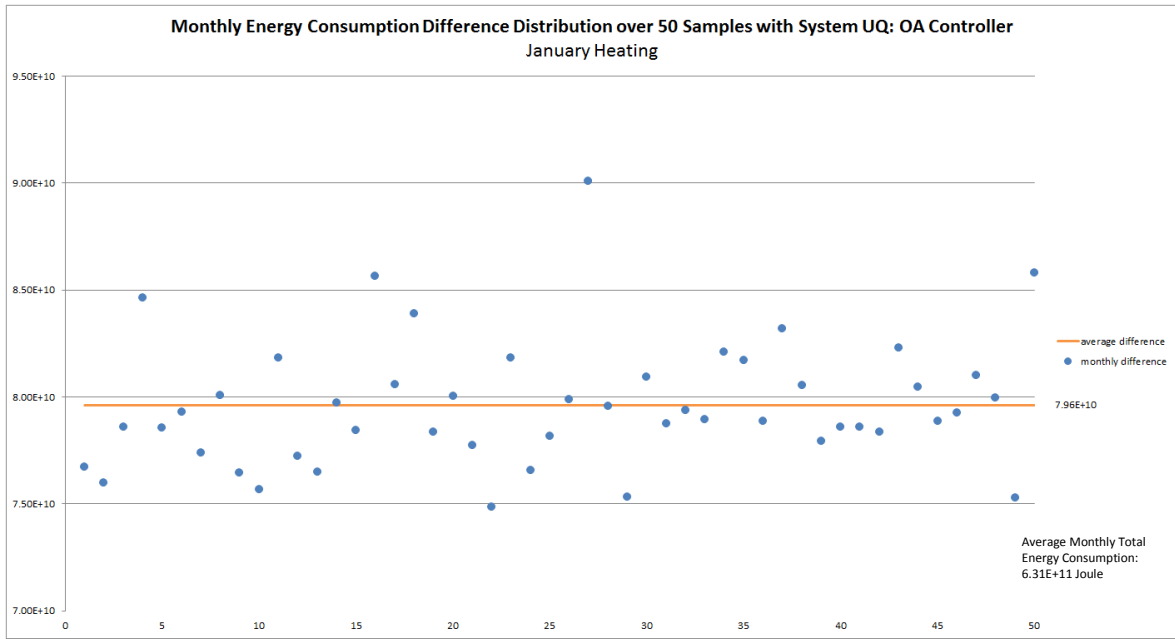


Figure 32 Monthly Energy Consumption Difference Distribution over 50 Samples for System UQ: OA Controller, January

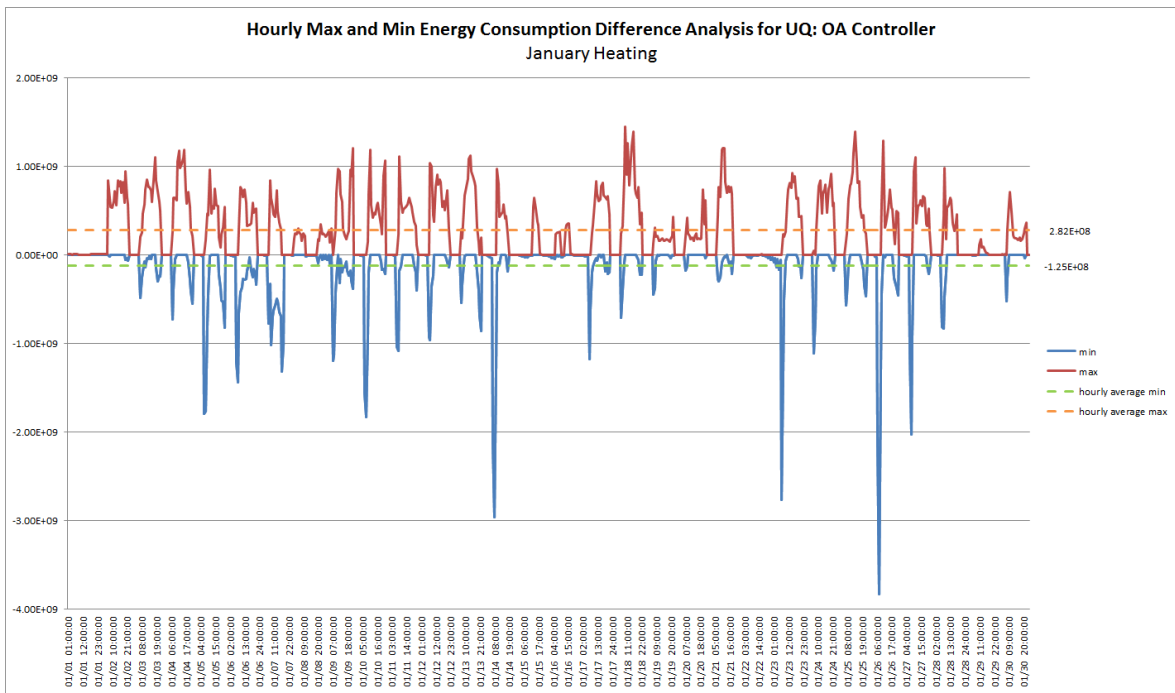


Figure 33 Hourly Max and Min Energy Consumption Difference Analysis for UQ: OA Controller, January

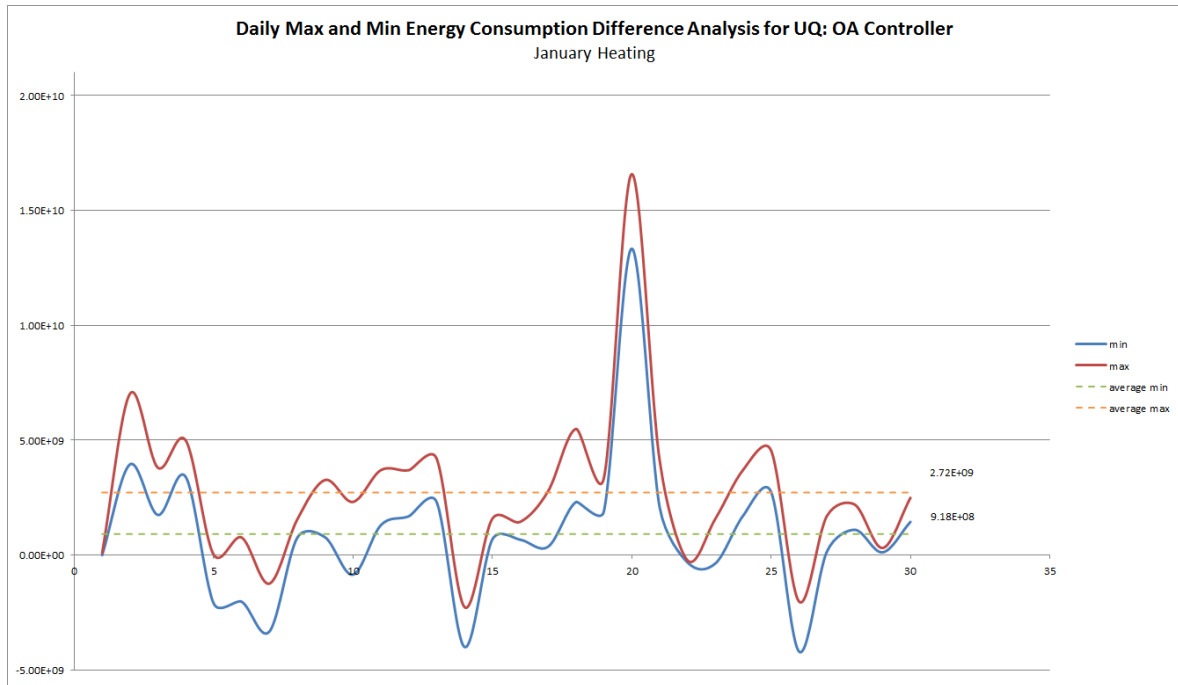


Figure 34 Daily Max and Min Energy Consumption Difference Analysis for UQ: OA Controller, January

Table 2 Daily Average Energy Consumption with UA: OA Controller, January

Date	1	2	3	4	5	6	7	8	9	10
average energy consumption (J)	1.02E+08	1.33E+10	1.91E+10	1.66E+10	2.65E+10	3.25E+10	4.18E+10	5.93E+09	5.04E+10	3.15E+10
Date	11	12	13	14	15	16	17	18	19	20
average energy consumption (J)	2.05E+10	1.80E+10	1.50E+10	2.31E+10	1.99E+09	3.26E+09	2.27E+10	1.67E+10	2.35E+10	1.17E+11
Date	21	22	23	24	25	26	27	28	29	30
average energy consumption (J)	1.21E+10	4.63E+08	2.19E+10	1.26E+10	1.24E+10	2.47E+10	2.20E+10	1.60E+10	3.11E+08	8.39E+09

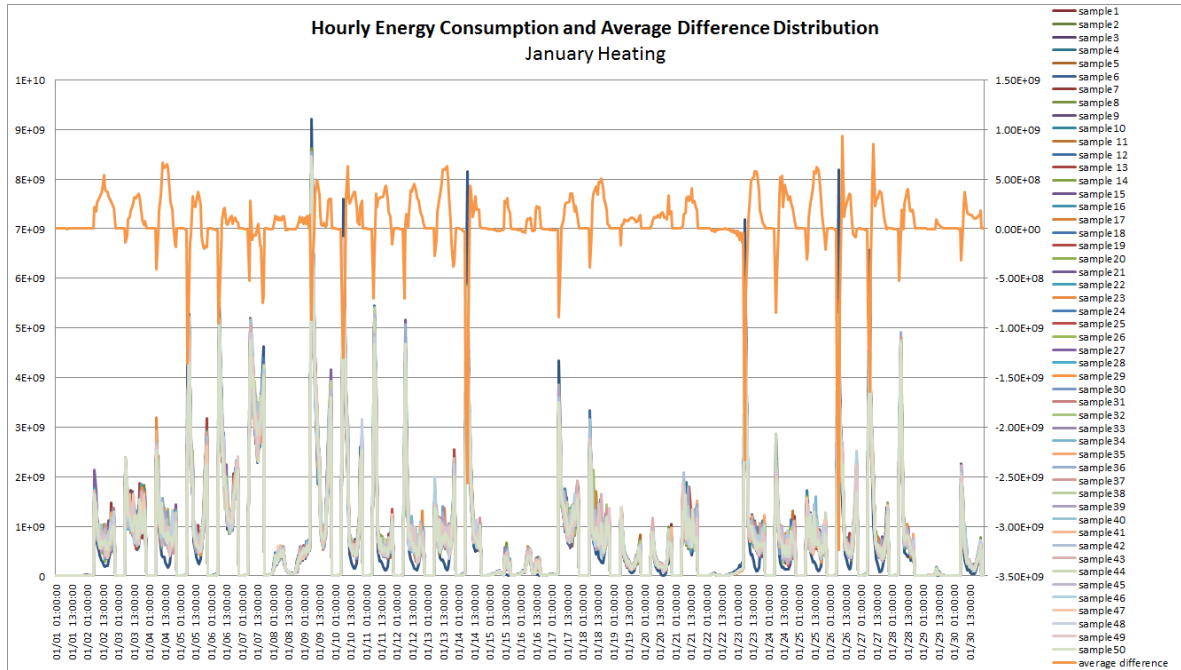


Figure 35 Hourly Energy Consumption and Hourly Average Energy Consumption Difference Distributions for UQ: OA Controller, January

Figure 31 shows that uncertainty of OA controller has respectable impacts on the energy consumption during winter, and most of the deviation is positive, which means that OA controller uncertainty mainly results in a noticeable raise in energy consumption in winter Atlanta. Those inferences can also be verified based in Figure 32. The monthly average energy consumption difference shown in Figure 32 is  $7.96E+10$ , which is positive and about 17.21% of the basement monthly energy consumption in January.

According to Figure 33 and Figure 34, the hourly energy consumption for January could vary from  $-1.25E+08$  (-19.44%) to  $2.82E+08$  (43.94%), and the daily energy consumption could vary from  $9.18E+08$  (4.39%) to  $2.72E+09$  (13.02%), if the system has an OA controller uncertainty range from 0.8 to 1.2.

Unlike UA uncertainty, Figure 35 seems to show a regularly repeated pattern of energy consumption difference during certain period of time. To look at the values in a more explicit way, 4 groups of values of hourly energy consumption difference are

shown as below in color scaled cells, in which red represents top 10% highest value, blue represents 10% lowest value, and yellow represents 0.



Figure 36 Hourly Energy Consumption Difference for January 4<sup>th</sup>

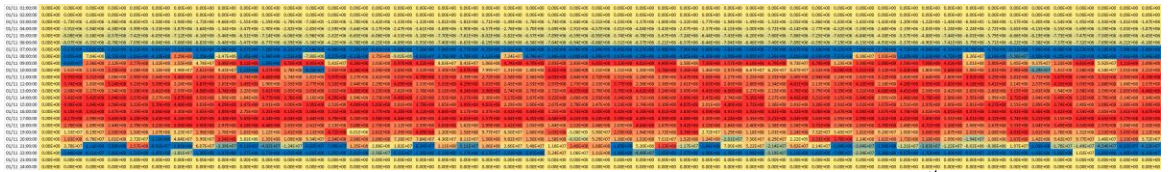


Figure 37 Hourly Energy Consumption Difference for January 11<sup>th</sup>



Figure 38 Hourly Energy Consumption Difference for January 18<sup>th</sup>



Figure 39 Hourly Energy Consumption Difference for January 25<sup>th</sup>

Based on Figure 36 to Figure 39, from 12:00:00 to 18:00:00 building has higher energy consumption then the baseline. For morning and night, the building usually has a lower energy consumption then the baseline with OA controller uncertainty. This pattern is reasonable since Atlanta has relative warm winter days, and the outdoor air temperature in the afternoon may meets the requirements and trigger the economizer to bring in more outdoor air, which gives OA controller more space to affect the energy consumption.



b. July

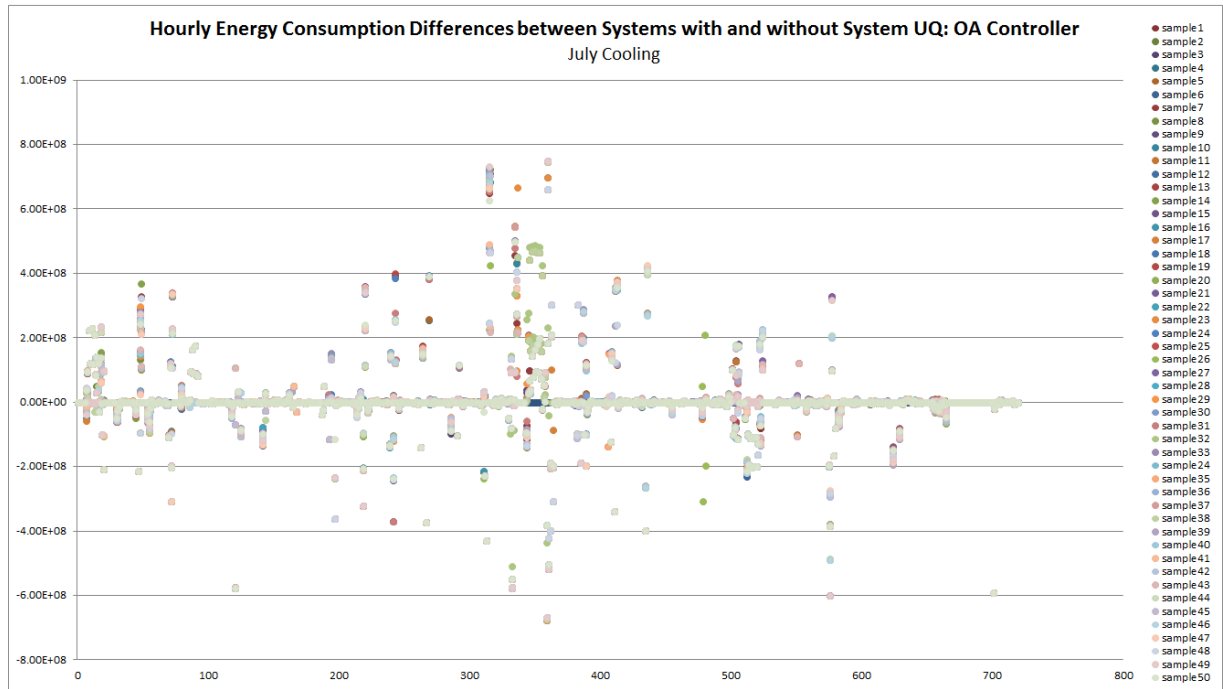


Figure 40 Hourly Energy Consumption Differences between Systems with and without System UQ: OA Controller, July

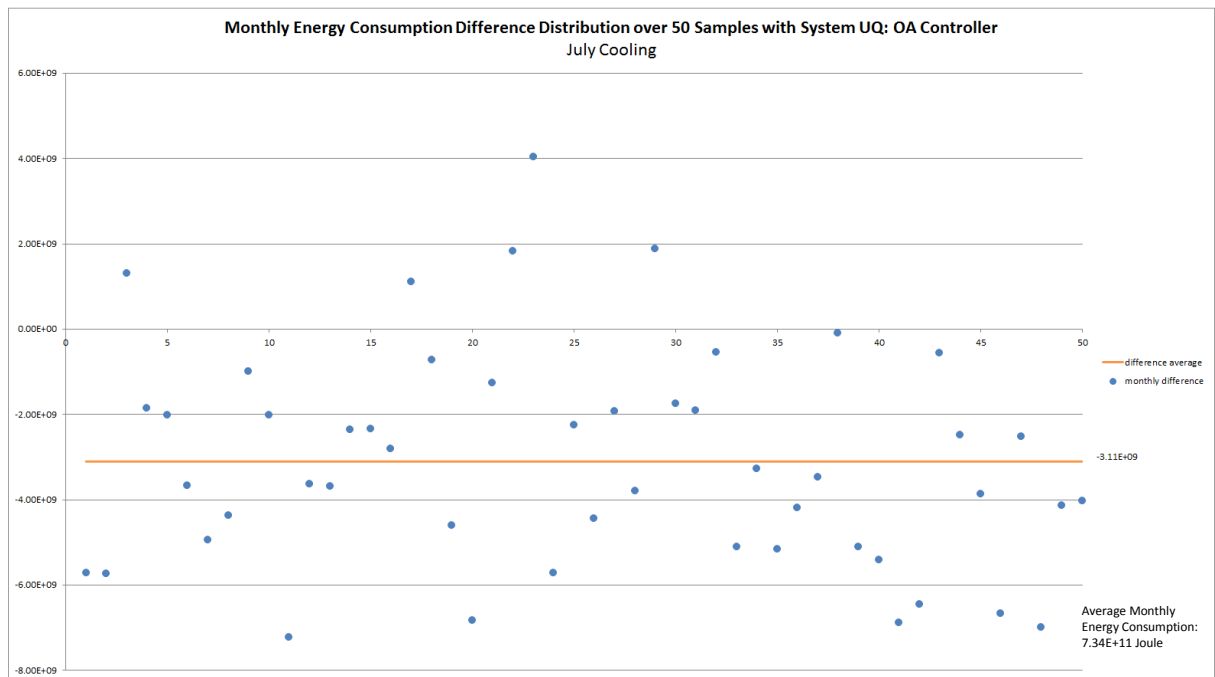


Figure 41 Monthly Energy Consumption Difference Distribution over 50 Samples for System UQ: OA Controller, July

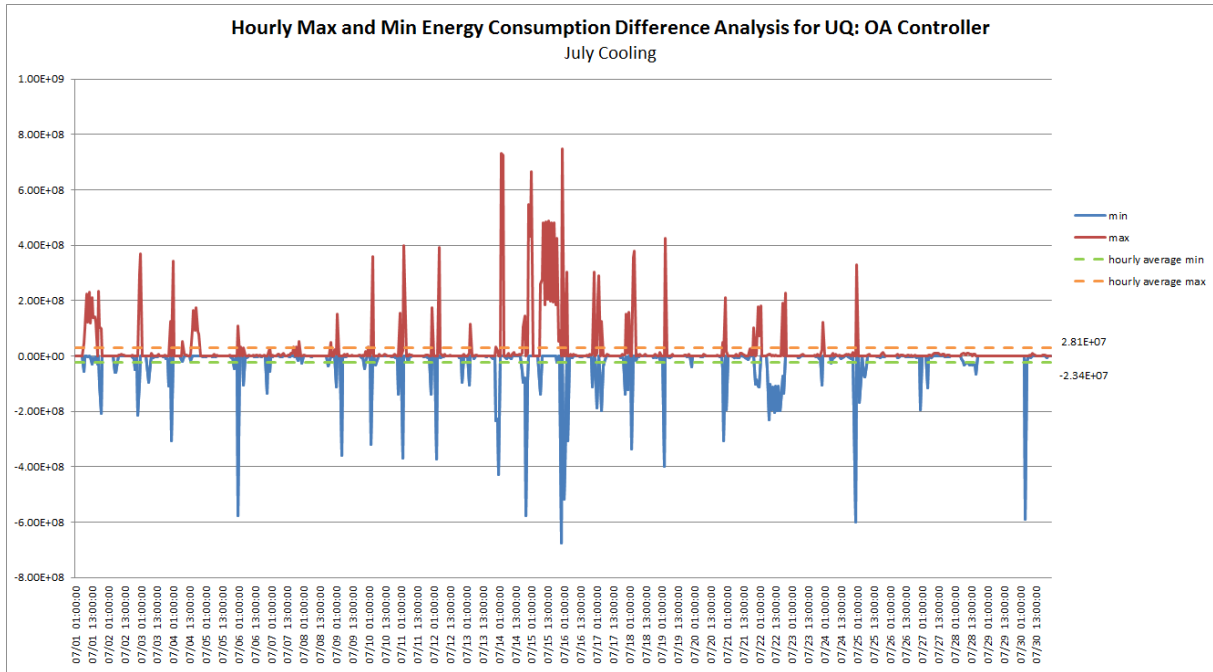


Figure 42 Hourly Max and Min Energy Consumption Difference Analysis for UQ: OA Controller, July

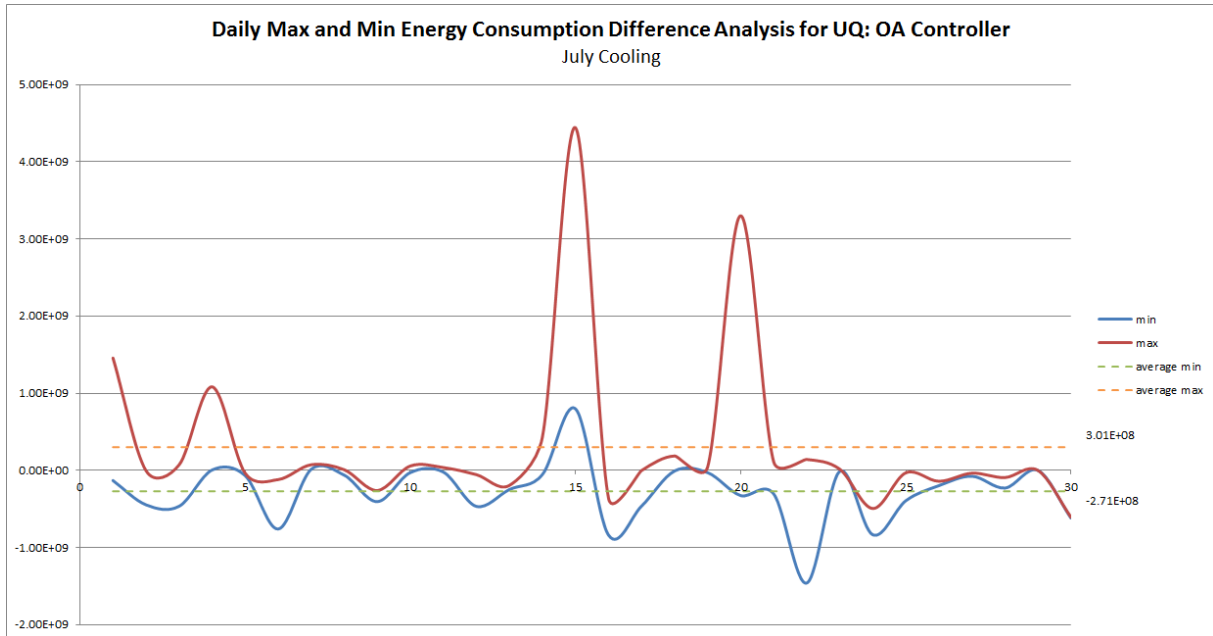


Figure 43 Daily Max and Min Energy Consumption Difference Analysis for UQ: OA Controller, July

Table 3 Daily Average Energy Consumption with UA: OA Controller, July

Date	1	2	3	4	5	6	7	8	9	10
average energy consumption (J)	1.32E+10	1.94E+10	1.95E+10	1.79E+10	2.12E+10	2.30E+10	2.15E+10	2.03E+10	2.11E+10	2.00E+10
Date	11	12	13	14	15	16	17	18	19	20
average energy consumption (J)	1.98E+10	2.08E+10	2.09E+10	1.95E+10	1.80E+10	2.20E+10	1.99E+10	2.01E+10	1.98E+10	1.88E+11
Date	21	22	23	24	25	26	27	28	29	30
average energy consumption (J)	2.10E+10	1.54E+10	1.98E+10	1.93E+10	1.91E+10	2.06E+10	1.93E+10	1.38E+10	0.00E+00	1.97E+10



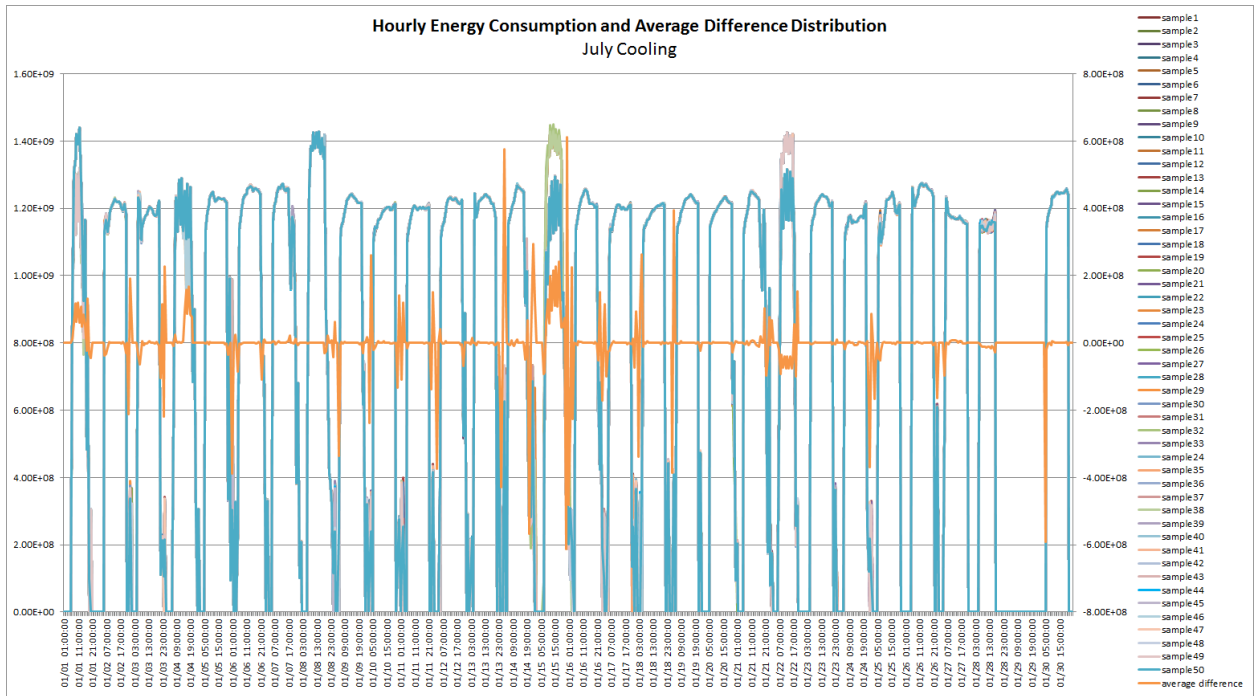


Figure 44 Hourly Energy Consumption and Hourly Average Energy Consumption Difference Distributions for UQ: OA Controller, July

Figure 40 and Figure 41 shows that uncertainty of OA controller in summer doesn't have significant impacts on the energy consumption. The monthly average energy consumption difference shown in Figure 41 is  $-3.11\text{E}+9$ , which is negative and about -0.55% of the basement monthly energy consumption in July.

Figure 42 and Figure 43 also show relatively small impacts on hourly and daily energy consumptions compared to the ones in January. By comparing to baseline hourly and daily average energy consumption, the deviations caused by uncertainty in OA controller range from -2.96% to 3.56% for hourly energy consumption, and from -1.29% to 1.44% for daily.

Same as the one in January, it's easy to tell from Figure 44 that most of the large deviations appear when hourly energy consumptions are low, which is in morning or at night, and most deviations are negative. One possible reason is that since Atlanta has very hot summer, outdoor air in the morning or near at night will be more suitable for

free cooling, and extra outdoor air brought in by OA controller uncertainty can result in energy consumption decrease.

### Water and air distribution

We test the uncertainty of water distribution and supply air distribution together because they both take place in the same module in EnergyPlus. We run 33 groups of samples for January and 30 groups of samples for July. Following are the results.

#### a. January

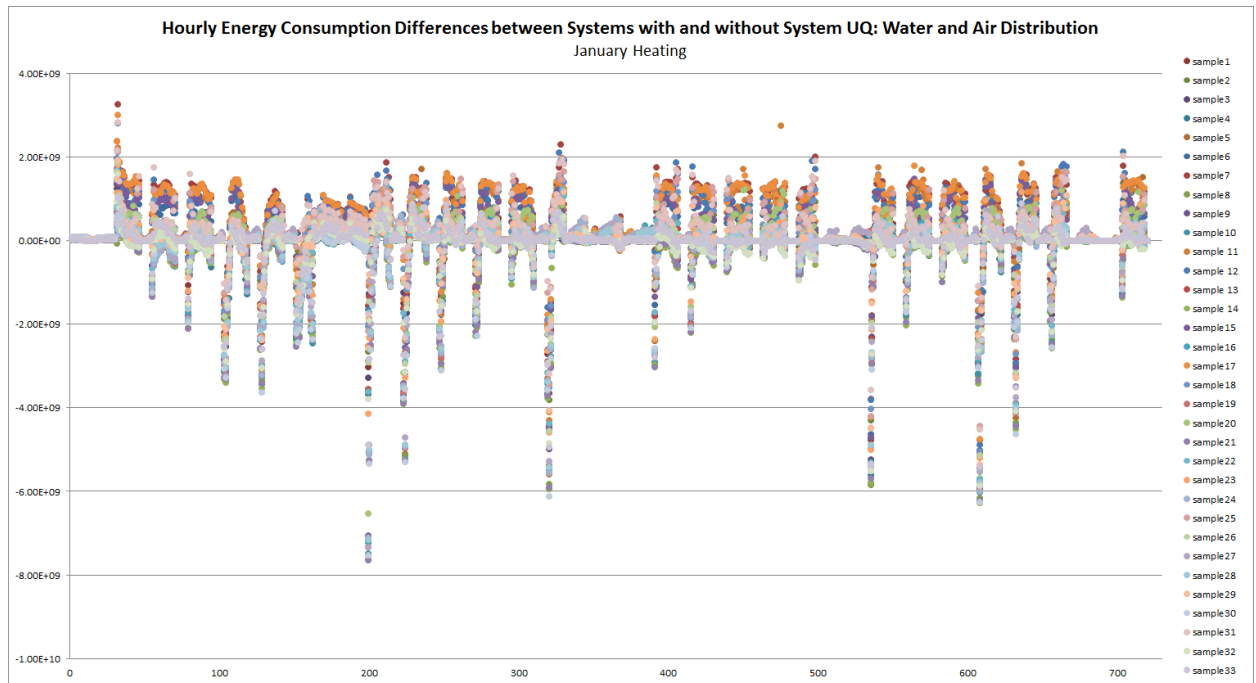


Figure 45 Hourly Energy Consumption Differences between Systems with and without System UQ: Water and Air Distribution, January

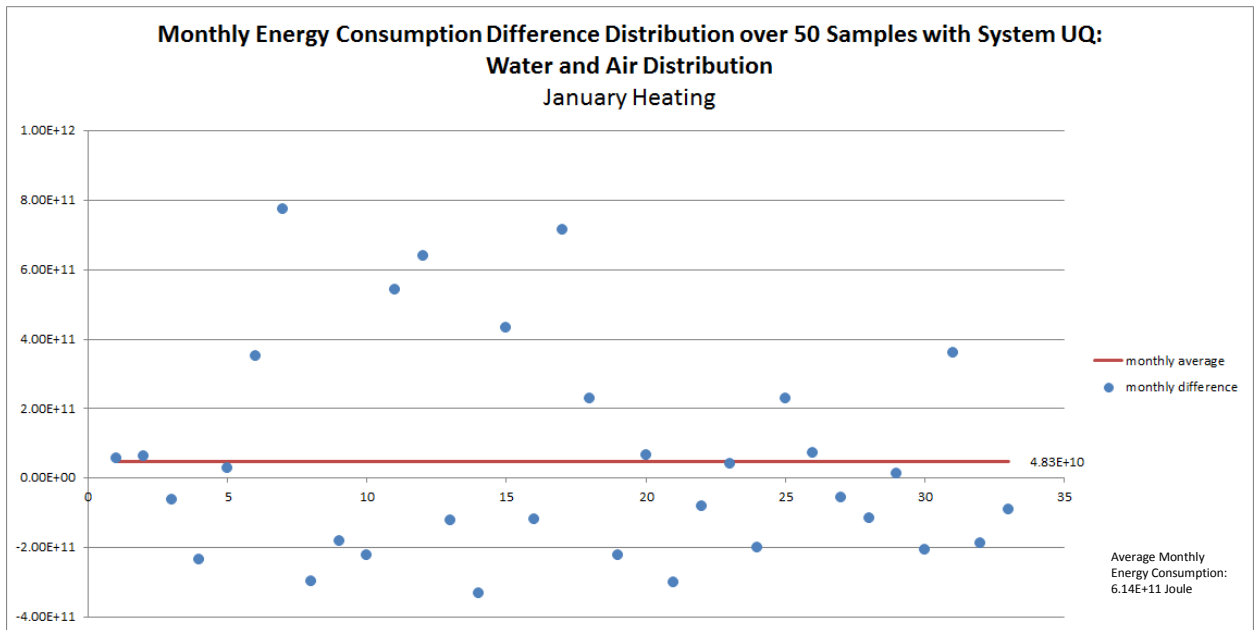


Figure 46 Monthly Energy Consumption Difference Distribution over 50 Samples for System UQ: Water and Air Distribution, January

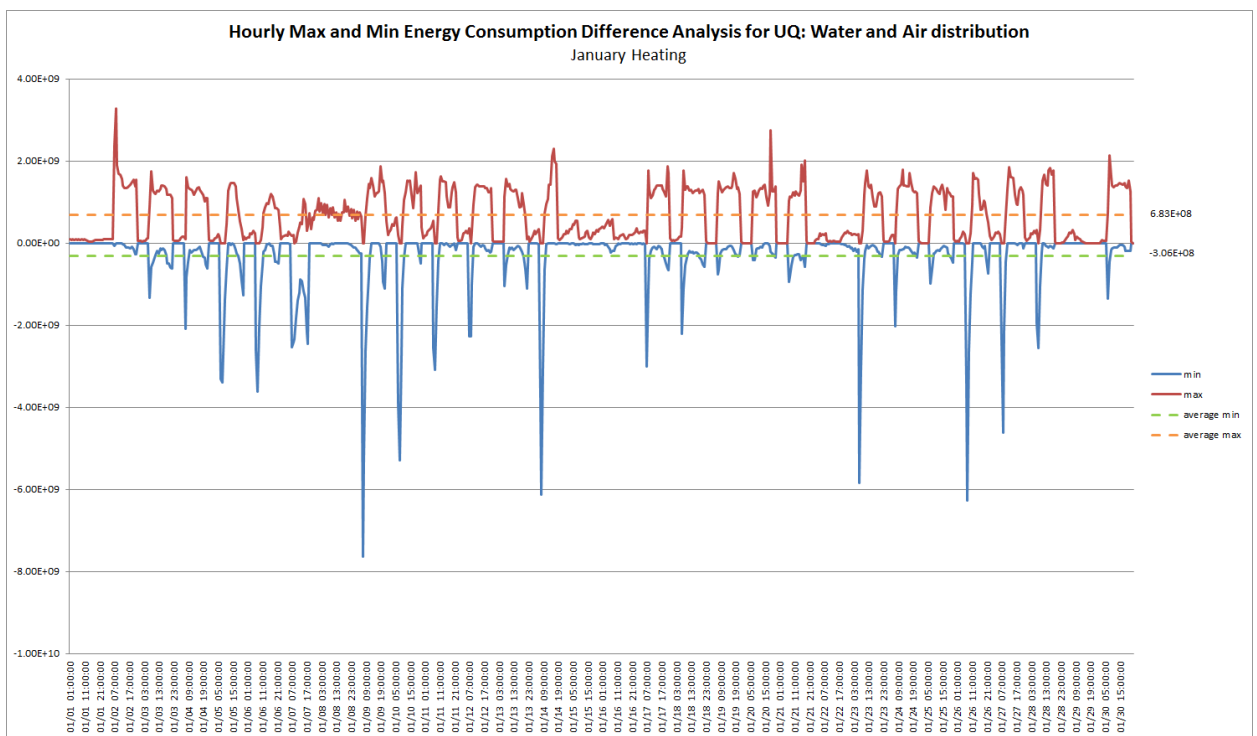
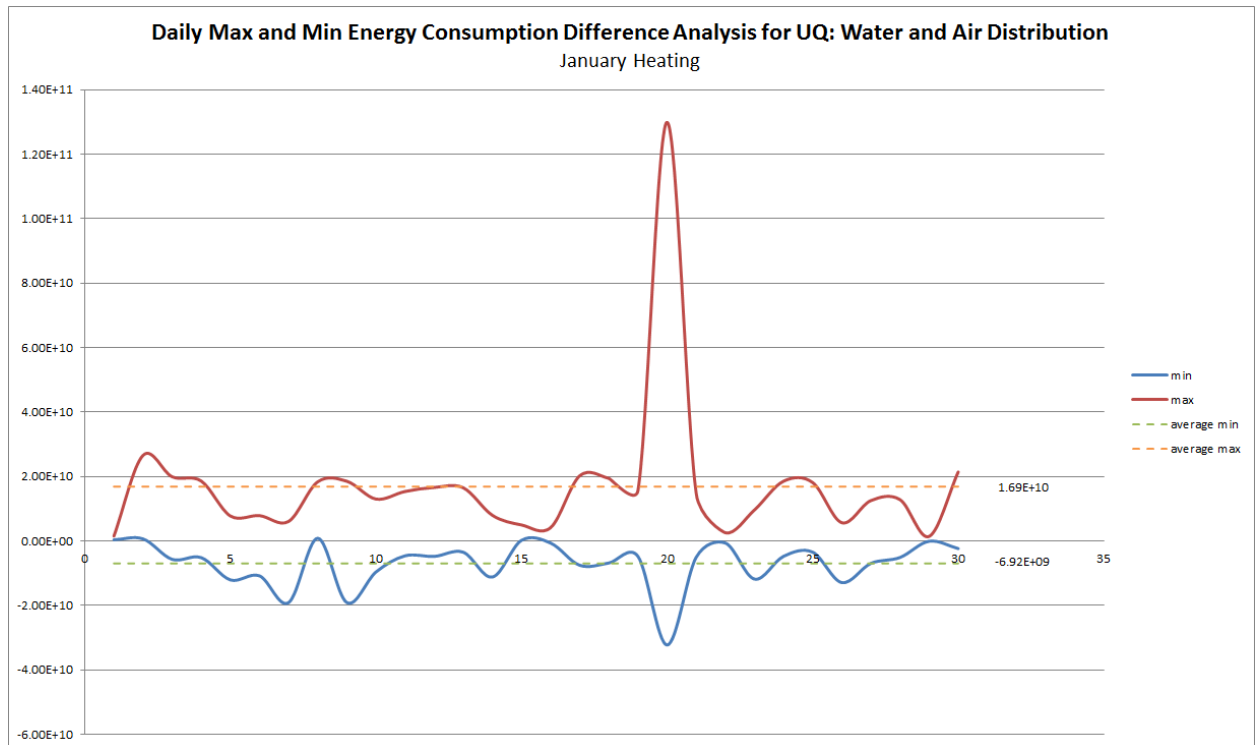


Figure 47 Hourly Max and Min Energy Consumption Difference Analysis for UQ: Water and Air distribution, January



*Figure 48 Daily Max and Min Energy Consumption Difference Analysis for UQ: Water and Air Distribution, January*

*Table 4 Daily Average Energy Consumption with UA: Water and Air Distribution, January*

Date	1	2	3	4	5	6	7	8	9	10
average energy consumption (J)	8.98E+08	1.72E+10	1.95E+10	1.45E+10	2.22E+10	3.00E+10	3.60E+10	1.17E+10	4.53E+10	2.95E+10
Date	11	12	13	14	15	16	17	18	19	20
average energy consumption (J)	2.01E+10	1.73E+10	1.43E+10	2.29E+10	2.53E+09	3.50E+09	2.41E+10	1.45E+10	2.32E+10	1.22E+11
Date	21	22	23	24	25	26	27	28	29	30
average energy consumption (J)	1.10E+10	1.05E+09	1.69E+10	1.24E+10	1.18E+10	2.15E+10	2.10E+10	1.60E+10	4.84E+08	1.13E+10

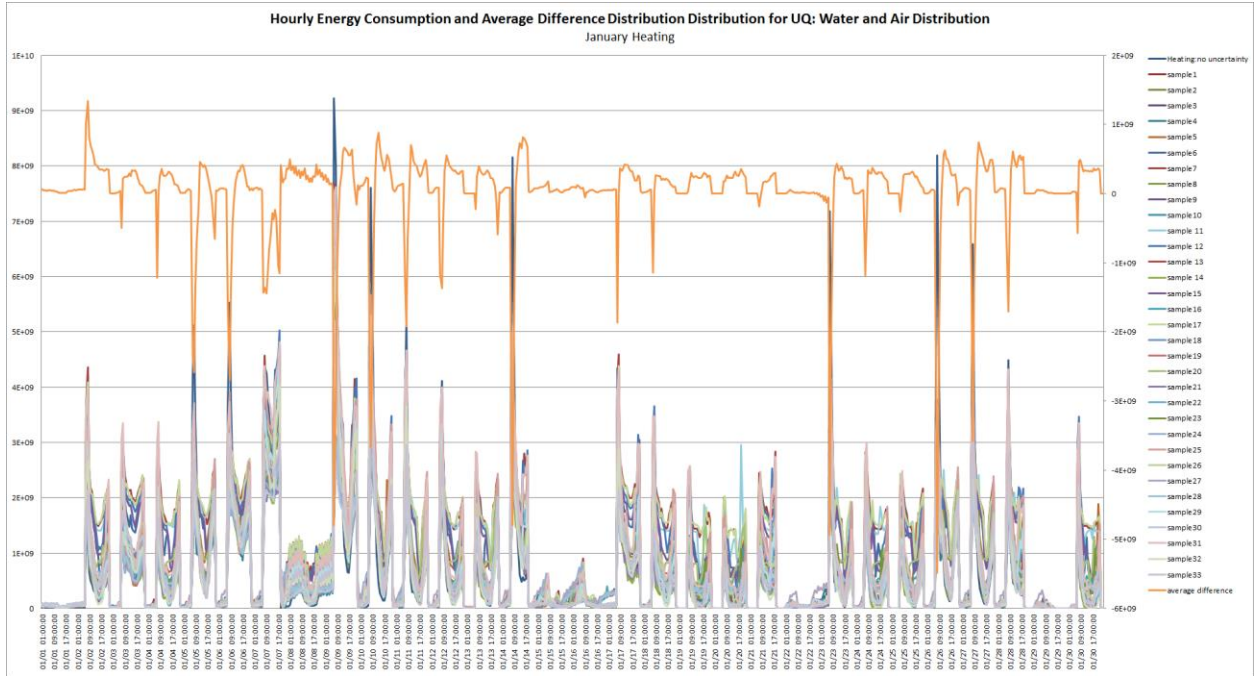


Figure 49 Hourly Energy Consumption and Hourly Average Energy Consumption Difference Distributions for UQ: Water and Air Distribution, January

Although the absolute values of negative deviations showed in Figure 45 are mostly larger than ones of positive deviations, positive deviation has a large number of data sample quantity. As a result, the monthly average energy consumption difference calculated based on Figure 46 is  $4.83\text{E}+10$  (10.44% compare to baseline).

According to Figure 47 and Figure 48, the hourly energy consumption of the building in January could deviate from  $-3.06\text{E}+08$  (-47.68%) to  $6.83\text{E}+08$  (106.30%), and the daily deviations could vary from  $-6.92\text{E}+09$  (-33.06%) to  $1.69\text{E}+10$  (80.82%).

Same as energy consumption with OA controller uncertainty in January, Figure 49 shows a regularly repeated pattern of hourly energy consumption difference, and the trend of average hourly energy consumption difference is similar to the trend of energy consumption distribution, which makes sense cause when the supply air volume and supply water flow rate go up, the two networks get more potential to be more unbalanced, which results in larger energy consumptions.

b. July

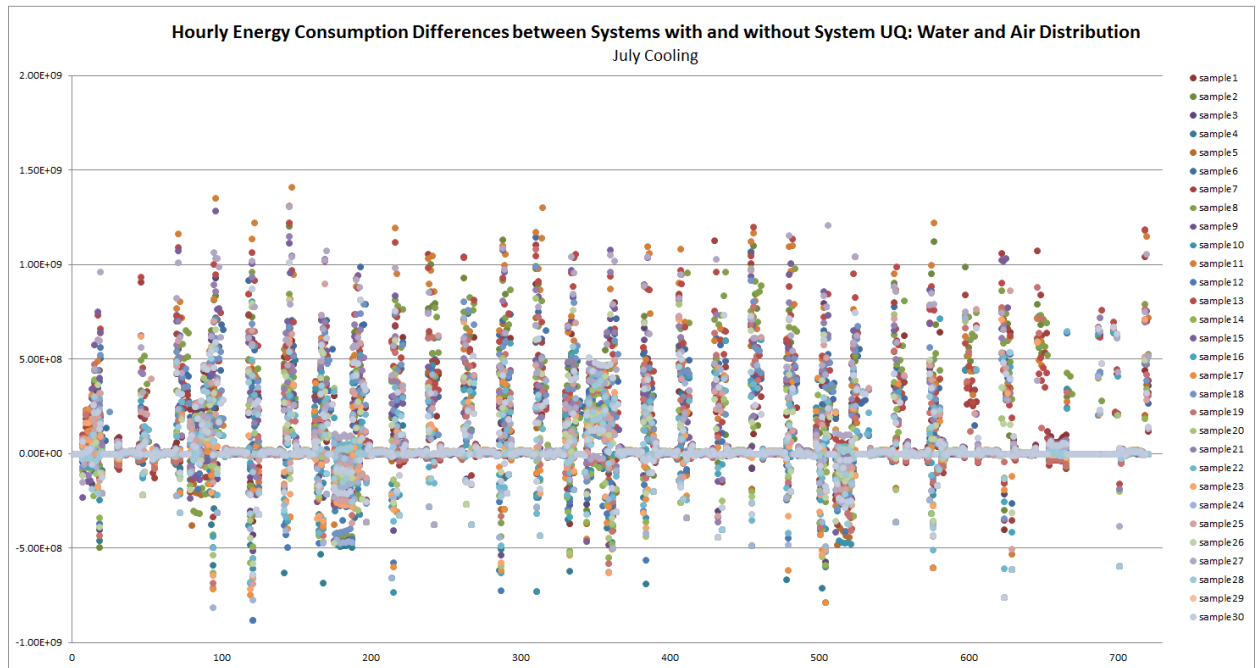


Figure 50 Hourly Energy Consumption Differences between Systems with and without System UQ: Water and Air Distribution, July

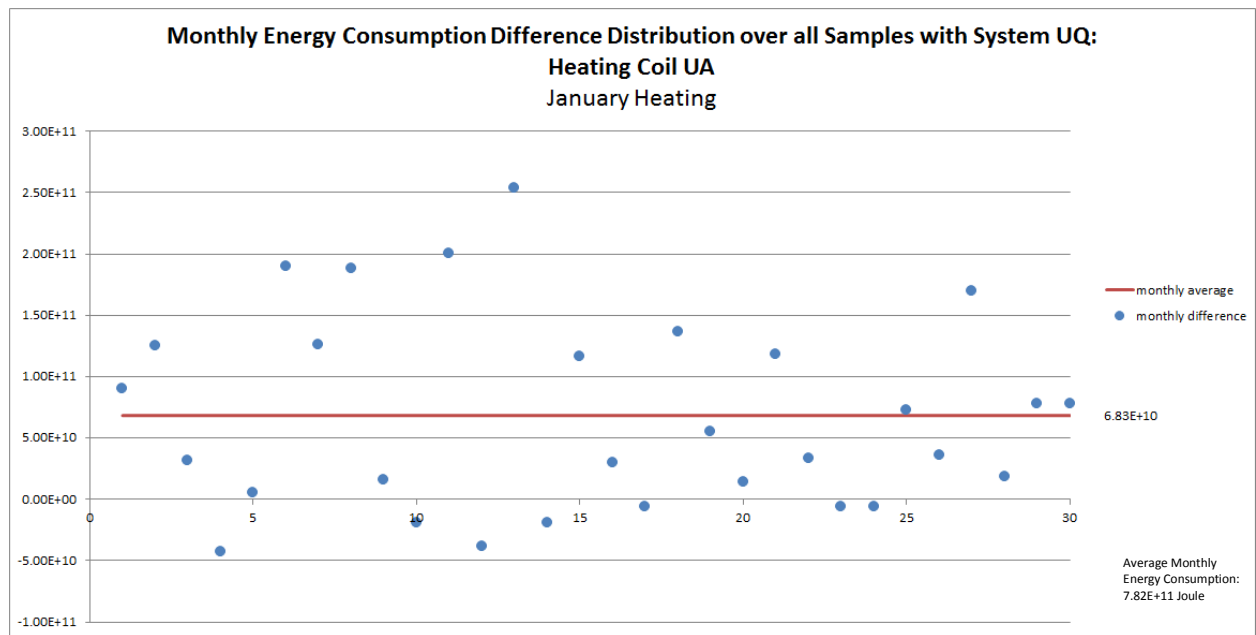


Figure 51 Monthly Energy Consumption Difference Distribution over 50 Samples for System UQ: Water and Air Distribution, July

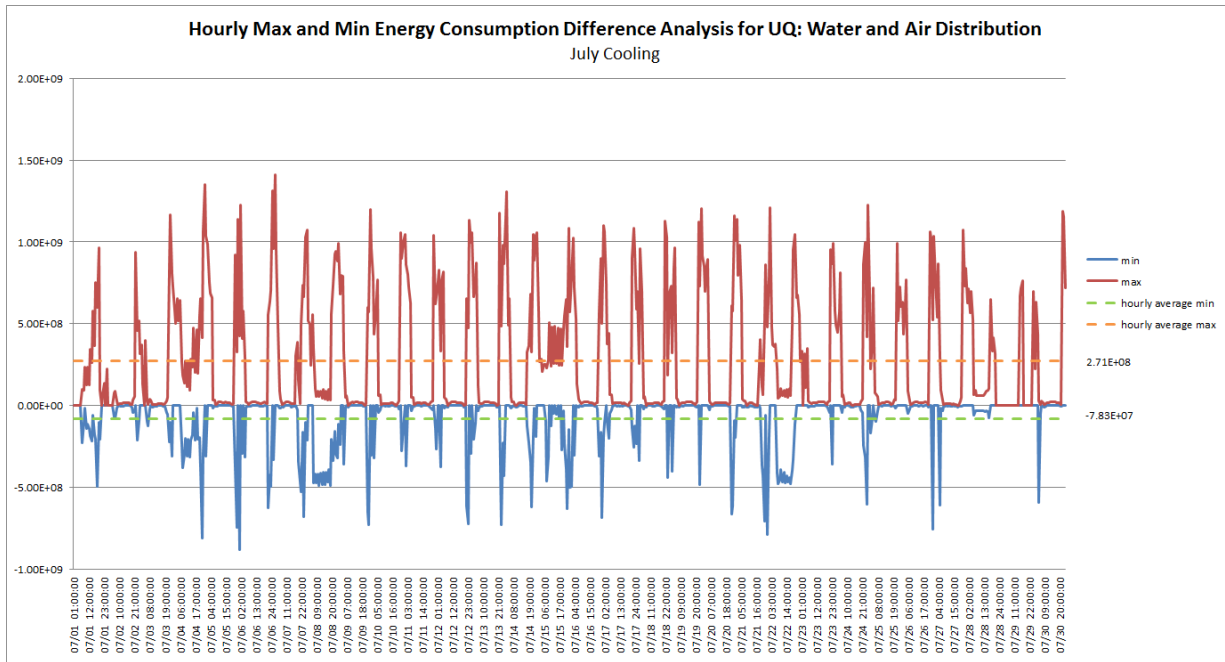


Figure 52 Hourly Max and Min Energy Consumption Difference Analysis for UQ: Water and Air Distribution, July

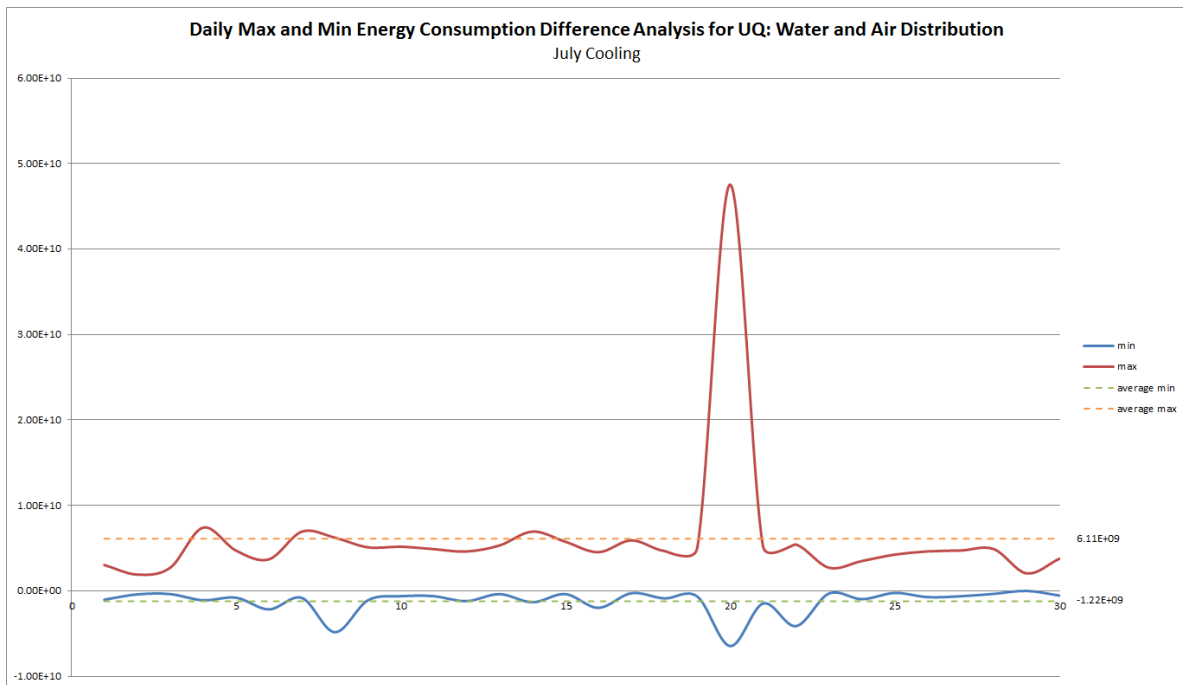


Figure 53 Daily Max and Min Energy Consumption Difference Analysis for UQ: Water and Air Distribution, July

Table 5 Daily Average Energy Consumption with UA: Water and Air Distribution, July

Date	1	2	3	4	5	6	7	8	9	10
average energy consumption (J)	1.35E+10	1.98E+10	2.06E+10	1.98E+10	2.27E+10	2.39E+10	2.36E+10	2.03E+10	2.25E+10	2.10E+10
Date	11	12	13	14	15	16	17	18	19	20
average energy consumption (J)	2.08E+10	2.24E+10	2.29E+10	2.11E+10	1.98E+10	2.35E+10	2.17E+10	2.07E+10	2.10E+10	2.03E+11
Date	21	22	23	24	25	26	27	28	29	30
average energy consumption (J)	2.18E+10	1.61E+10	2.06E+10	2.06E+10	2.04E+10	2.16E+10	2.02E+10	1.45E+10	5.72E+08	2.13E+10

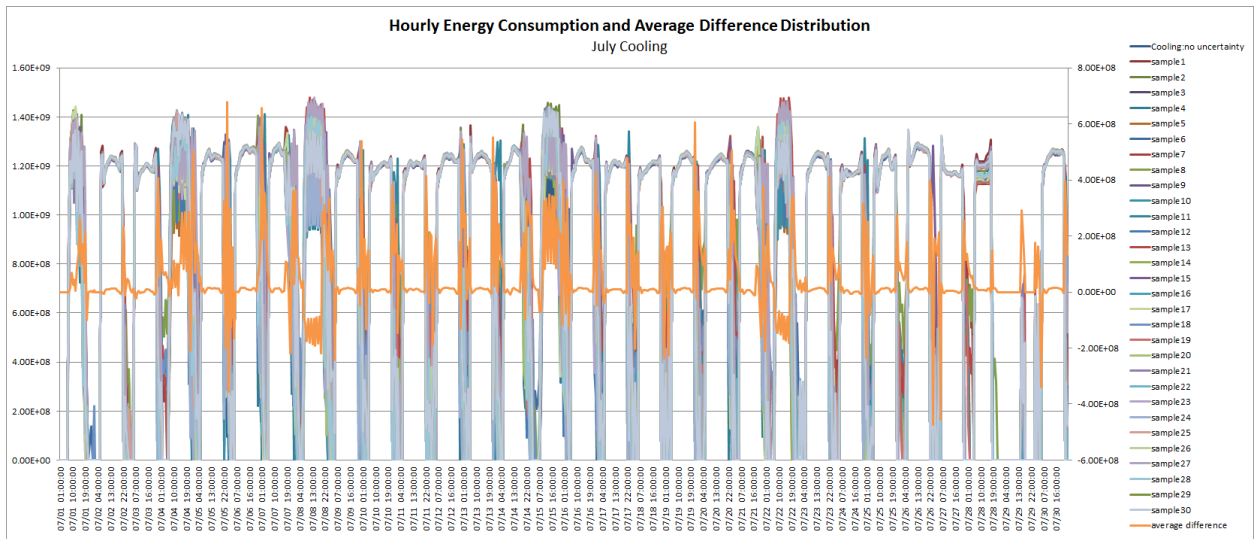


Figure 54 Hourly Energy Consumption and Hourly Average Energy Consumption Difference Distributions for UQ: Water and Air Distribution, July

Based on Figure 51, Figure 52, Figure 53 and the baseline value for monthly, hourly, daily average energy consumption, the monthly average energy consumption difference is  $6.83\text{E}+10$  (12.02%), the hourly energy consumption could deviate from  $-7.83\text{E}+07$  (-12.18%) to  $2.71\text{E}+08$  (42.18%), and the daily deviations could vary from  $-1.22\text{E}+09$  (-5.84%) to  $4.68\text{E}+09$  (22.37%).

According to Figure 54, the trend of average hourly energy consumption difference is similar with but not exactly follows the trend of energy consumption distribution. One possible reason is that the reheat coils will not be using that much in July as they are in January, thus the water distribution unbalance will not be a major factor that affect the water distribution uncertainty. Without water distribution unbalancing problem, the water distribution uncertainty will be trivial.



## CHAPTER 7

### CONCLUSION AND FUTURE WORK

This study investigates the features of 5 HVAC operation uncertainties detected in the case building, generates methods that can capture the impacts of those uncertainties, upgrades EnergyPlus by updating its source code, eventually testing the impacts of the 5 uncertainties using GURA-W by UA based on sampling over a hypothetical uncertainty range which is taken as  $\pm 20\%$ , using the extended version of EnergyPlus with upgraded AHU modules.

The impact of each uncertainty is shown below. Red represents top 10% highest value, blue represents 10% lowest value, and yellow represents 0.

*Table 6 Impact of each uncertainty to hourly, daily and monthly energy consumptions*

energy consumption difference (%)	January					July				
	hourly		daily		monthly	hourly		daily		monthly
	min	max	min	max	average	min	max	min	max	average
UA Factor	-0.002%	0.002%	-0.001%	0.001%	<0.001%					
Temperature Sensor					<0.001%					
OA Controller	19.440%	43.94%	4.39%	13.02%	17.21%	-2.96%	3.50%	-1.29%	1.44%	0.55%
Water and Air Distributions	-47.68%	106.30%	-33.06%	80.82%	10.44%	-12.18%	42.18%	-5.84%	22.37%	12.02%

Since all the uncertainties are simulated under the same range, it is safe to say that the impacts on energy consumption by uncertainty in heating coil UA and temperature sensors are very small. Deviation introduced by uncertainty in OA controller and in water and air distributions can be significantly higher.

The next step of this work is to apply those system operation uncertainties to real simulations. But some issues deserve special attention:

#### **Limitations:**

For the heating coil UA factor, our treatment has only shown how to add UA factor uncertainty to heating coils and not cooling coils. As cooling coil play an important role

in cooling mode, it is necessary to also investigate how EnergyPlus models cooling coil, and add UA uncertainty to it.

For temperature sensor uncertainty, it would be better to relocate the sensor uncertainty to zone temperature sensor, as we addressed in the last chapter.

For water distribution, as it was introduced in the last chapter, our upgraded EnergyPlus model cannot model the unbalancing of AHU coils and VAV reheat coils together (as addressed at the end of the last chapter). To connect those two types of coils together, one needs to loop over Manage HVAC twice and make sure nothing else will be affected, or find another way to capture the uncertainty.

We only considered 5 uncertainties, while there still a lot of additional operation uncertainties exist in the system, for instance, the uncertainty of supply air leakage, and uncertainties caused by boiler and chiller operation.

### ***Applicability to other buildings***

Since the uncertainties are caused by HVAC operation, which are specific to the HVAC system, the specifics of the building will not have a significant influence factor on the operation uncertainty. Thus once the uncertainties have been well captured, we can use them in the simulation of any building with the same HVAC system type.

### ***Future work: Quantification of the uncertainty ranges in uncertainty analysis of building designs***

As the uncertainties are mainly affected by HVAC system type, we can collect data from a set of monitored buildings with the same HVAC system type, and calibrate the uncertainty ranges over the set of buildings. This requires selection of suitable buildings with appropriate metered data.

The ultimate objective of the work will be to do a full uncertainty analysis of building energy models and determine through sensitivity analysis what the role is of the system parameters on the resulting energy performance. This will lead to generic statements about the need to consider system uncertainty as part of a simulation, and the need to make system upgrades part of a planned retrofit.

## REFERENCES

- L. Wang, P. Mathew, X. Pang (2012). *Uncertainties in energy consumption introduced by building operations and weather for a medium-size office building*, Energy and Buildings, 2012.
- B. Yan and A., Malkawi (2012). *Predicting System Performance with Uncertainty*, In Proceedings of the Twelfth International Conference for Enhanced Building Operations (ICEBO) Held in Manchester, the United Kingdom, 23-26, October 2012. Manchester, the United Kingdom.
- M. Basarkar, X. Pang, L. Wang, P. Haves, T. Hong (2011). *Modeling and Simulation of HVAC Faults in EnergyPlus*, In Proceeding of Building Simulation 2011: 12<sup>th</sup> Conference of International Building Performance Simulation Association, Sydney, 14-16 November.
- EnergyPlus (2011). *Guide for Module Developers*, Orlando: Ernest Orlando Lawrence Berkeley National Laboratory.
- EnergyPlus (2011). *EnergyPlus Engineering Reference*, Orlando: Ernest Orlando Lawrence Berkeley National Laboratory.
- X. Zhang (2001). *Heat Transfer (4<sup>th</sup> Edition)*. Beijing: China Architecture and Building Press, 2001. Print.
- B.D. Lee, Y. Sun, G. Augenbroe, C.J.J. Paredis (2013). *Toward Better Prediction of Building Performance: A Workbench to Analyze Uncertainty in Building Simulation*, in, 13th International Building Performance Simulation Association Conference (To be Published), 2013.